



# Documentation Update for *Siebel Reports Administration Guide, MidMarket Edition*

**Date Published:** December 28, 2001

**Last Updated:** December 28, 2001

This Documentation Update applies to the following version of *Siebel Reports Administration Guide, MidMarket Edition*.

**Document Version:** 7.0

**Date Published:** September 2001

**Software Version:** Siebel 7 v7.0.2

# General Information

December 28, 2001

## List of Reports in Siebel 7 with Navigational Path

The reports included in this list are those available across various modules in Siebel 7. Some of these reports may be removed and new reports added in later versions of Siebel 7. Also, the functionality of the reports listed may be modified in future releases.

Template Name refers to the report design file name in ..\Tools\rptsrc\enu\standard. Parameterized reports are clearly marked.

Table 1-1 contains a listing of reports in the Siebel eBusiness Application, MidMarket Edition.

**Table 1-1. Listing of Reports in Siebel eBusiness Application, MidMarket Edition (1 of 2)**

Report Name	Template Name	Menu Text	Description	Navigation Path
Expected Revenue Forecast (SWG)	EXREVFOR	Forecast - Expected Revenue	Expected revenue forecast	
Forecast Product Revenue (SWG)	PRODFOR	Forecast - Product Revenue	Forecasted product revenue	
Revenue Forecast (SWG)	REVFOR	Forecast - Revenue	Revenue forecast spreadsheet	
Team Forecast Summary - Expected Product Revenue (SWG)	TMEXPRODFOR SUM	Team Forecast Summary - Expected Product Revenue	Summary of team's forecasted expected product revenue	
Team Expected Revenue Forecast (SWG)	TMEXREVFOR	Team Forecast - Expected Revenue	Team's forecasted expected revenue	
Team Expected Revenue Forecast Summary (SWG)	TMEXREVFORSUM	Team Forecast Summary - Expected Revenue	Summary of team's forecasted expected revenue	
Team Forecast Summary - Product Revenue (SWG)	TMPRODFORSUM	Team Forecast Summary - Product Revenue	Summary of team's forecast product revenue	
Team Revenue Forecast (SWG)	TMREVFOR	Team Forecast - Revenue	Team's forecasted revenue	

**Table 1-1. Listing of Reports in Siebel eBusiness Application, MidMarket Edition (2 of 2)**

<b>Report Name</b>	<b>Template Name</b>	<b>Menu Text</b>	<b>Description</b>	<b>Navigation Path</b>
Team Revenue Forecast Summary (SWG)	TMREVFORSUM	Team Forecast Summary - Revenue	Summary of team's forecasted revenue	
Team Forecast Product Revenue (SWG)	TPRODFOR	Team Forecast - Product Revenue	Team's forecasted product revenue	
Forecast Product Expected Revenue (SWG)	EPRODFOR	Forecast - Expected Product Revenue	Forecasted product revenue for a sales representative	
Team Forecast Expected Product Revenue (SWG)	ETPRODFOR	Team Forecast - Expected Product Revenue	Forecast product revenue for the team	

*The following section, which is intended for end users, was not included with Siebel Reports Administration Guide:*

## Reporting in the Web Client and the Dedicated Web Client

### About This Chapter

This chapter describes the user interaction with the Reports Server product in the Web Client and Dedicated Web Client (both connected and disconnected modes).

Siebel Reports Server is an out-of-the box integration of Actuate e.Reporting Server. While the end user interaction with Siebel Reports Server will be completely within Siebel, the administrator still needs to administer the reports encyclopedia using Actuate Administrator Desktop. In order for the users to run reports in the Web Client and connected Dedicated Web Client, the administrator should have installed Siebel Reports Server as described in *Siebel Server Installation Guide for Microsoft Windows* or *Siebel Server Installation Guide for UNIX*.

---

**NOTE:** In the disconnected mode of the Dedicated Web Client, reports can only be run interactively and hence the Reports Server views will not be available.

---

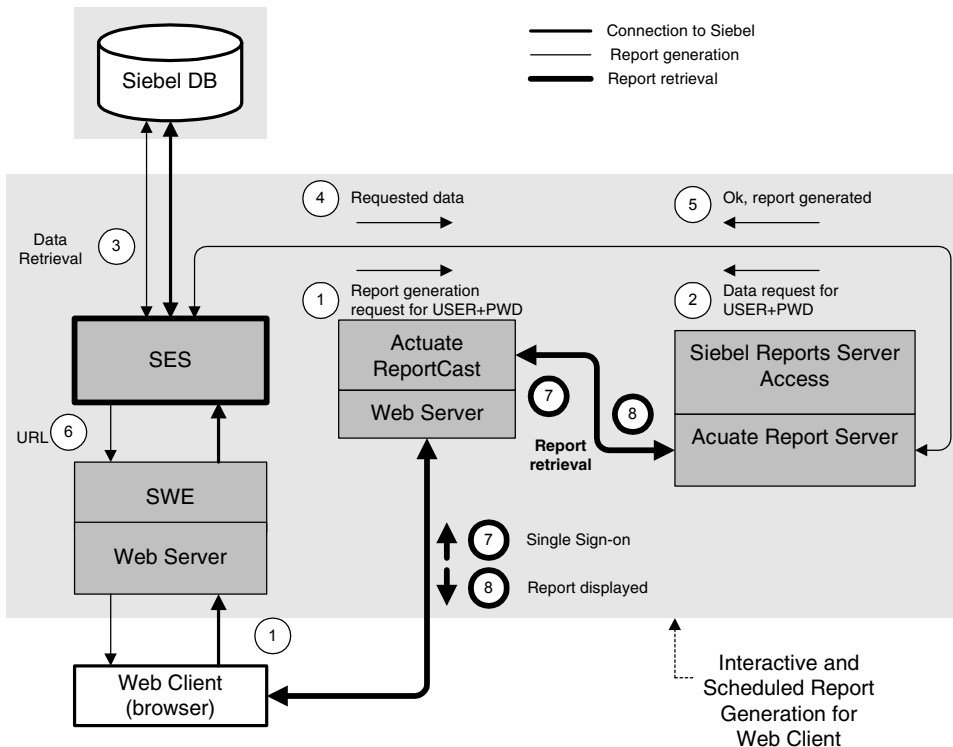
### Reporting in the Web Client

Siebel Reports Server allows Web Client users of Siebel eBusiness Applications to run reports both in interactive and batch modes. To run a report in a view, users first select from the application-level menu, View > Reports. The report window appears and allows the user to select a report from the drop-down list of reports available in that view. This dialog also allows the user to select the Run Now option to run the report interactively, or the Schedule option to run the report in batch mode at a later time. The Schedule option displays a separate dialog for the user to enter the schedule parameters.

## System Architecture

In this section, an overview of the Web Client report execution with Siebel Reports Server is described. The Reports Server encyclopedia consists of the complete set of report executables and the user folders. These report executables correspond to the complete set of available reports among all the views. These executables run in the Reports Server factory process (a multi-threaded report execution process) at report execution time and generate the report output file (ROI) by obtaining data from the Siebel Object Manager. The report output (ROI) file will be stored in the user folder in the reports encyclopedia and can be accessed from the Reports Server views in Siebel eBusiness Applications. The report execution is based on the parameter (ROV) file, which is generated by Siebel Object Manager when the user runs a report from a Siebel eBusiness Application view.

Figure 1-1 illustrates the report execution process from the Web Client.



**Figure 1-1. Server-Based Reporting for the Siebel Web Client**

The following is the sequence of steps corresponding to Figure 1-1:

- 1** Request submitted by the Web Client through SES (Siebel Enterprise Server).
- 2** Actuate e.Reporting Server runs the report and requests data from the SES through Siebel Reports Server Access.
- 3** Data retrieved by Siebel Object Manager from database.
- 4** Siebel Object Manager provides data to Reports Server.
- 5** Report generates and SES retrieves completed report status.
- 6** SES creates and passes the ReportCast report request to the Web Client.
- 7** Web Client connects to Actuate ReportCast, which authenticates user on the Reports Server.
- 8** Report in DHTML displays in the Web Client.

The report request submitted from the Web Client includes information regarding the current view, active query, sort specifications, and visibility rules. For all reports run in the batch mode (using Schedule option), the status of submitted report requests can be obtained from the Reports Server views.

---

**NOTE:** The user need not be logged into a Siebel eBusiness Application for the scheduled reports to be generated since report generation by the Reports Server is asynchronous. The data for report generation is obtained from the Siebel Object Manager running under the Siebel Server.

---

## User Interface and Process Flow

A user first chooses View > Reports and then selects the desired report to be run in the report window. Figure 1-2 illustrates the report window launched from the Opportunity views.



**Figure 1-2. Report Window View**

Then the user may indicate if the report should be run in interactive (Run Now) or batch (Schedule) mode:

- **Run Now:** If the user has selected Run Now, there are no further dialogs. After Run Now option is selected, the report will be displayed in the Web Client if it can be generated in 30 seconds or less, as defaulted in Run Report Time Limit parameter. The default value of this parameter can be changed in View > User Preferences > Behavior view. If the report generation takes longer than 30 seconds, the control is returned back to the Siebel Web Client and the message in [Figure 1-3](#) will be displayed. However, the report generation will continue in the Reports Server irrespective of the Run Report Time Limit parameter value. In the Dedicated Web Client, the Run Now option is the Run option.



**Figure 1-3. Run Report Time Limit Message**

---

**NOTE:** The browser frame displaying the report in DHTML will automatically close if the user navigates to another view in the application. If the user requires the report to be displayed when navigating to other views, they should select the save/print option in the viewer to display the report in PDF format.

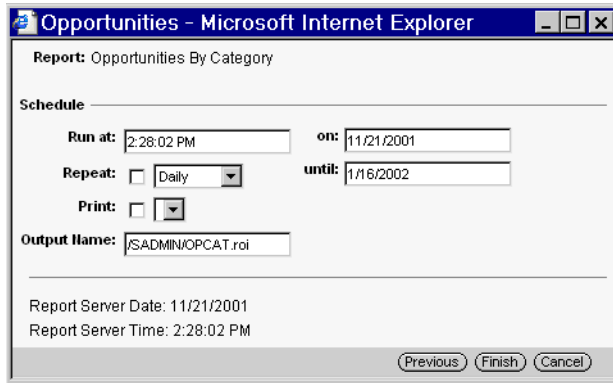
---

***To change the Run Report Time Limit parameter at the client level***

- 1 From the application-level menu, choose View > User Preferences.
- 2 Click the Behavior tab.
- 3 Enter a number in the Run Report Time Limit field to specify the number of seconds you are willing to wait to view the report.

The report will be generated in the Reports Server irrespective of the Run Report Time Limit parameter value.

- **Schedule:** In the Web Client, this selection indicates that you wish to run a report in batch mode by scheduling the report for one-time or periodic generation. If Schedule option is selected, the Schedule Report window appears, as shown below in [Figure 1-4](#).



**Figure 1-4. Schedule Report Window**

The Schedule Report window allows you to specify the time and date when the report is run, to select the option to generate the report with a recurring frequency, and to indicate if the report should be printed upon generation. You may also change the name of the report output file.

The fields of the Schedule Report dialog box are as follows:

- **Run at.** The time of day for report generation.
- **On.** For a one-time report, the date when the report is to be generated. For a periodic report, the first date the report is generated.
- **Repeat check box.** Specifies whether the report is one-time or periodic. Checked if periodic, unchecked if one-time.
- **Until.** Applies to periodic reports. This is the date after which the report is no longer run.
- **Print check box.** If this box is checked, the report is printed to the specified system printer upon its generation in the Reports Server.
- **Printer name picklist.** Allows the selection of a printer for report output on execution. The initially displayed printer name is the default printer specified in the user profile. The list of available printers is the set of printers administered for this purpose on the Reports Server.
- **Output Name.** Specifies the pathname in the Report Encyclopedia for the ROI file to be generated each time the report runs. This pathname is of the following form:

`/folder/reportname.roi`

where *folder* can contain subfolders if necessary. The default value in the Report field is your default Report Encyclopedia folder, as specified in the Personal Profile view, followed by the same report name as the executable. You can specify a destination folder other than your own default folder, but it must be present on the Report Encyclopedia, and you must have access to it through a user list.

- **Report Server Date.** Read-only field that indicates the equivalent Reports Server local date.
- **Report Server Time.** Read-only field that indicates the equivalent Reports Server local time.

After a report has been generated, the report output (ROI file) is available to the user for viewing and printing in the Completed Request Notifications, My Reports, or Explorer view. These are described in [“Siebel Reports Server Views” on page 1-12](#).

## Reporting in the Dedicated Web Client

Reports can be generated both in the connected and the disconnected mode of the Dedicated Web client. In the disconnected mode of the Dedicated Web Client, reports can only be generated interactively by accessing data from the local Object Manager. Further, interactively run reports (both in connected and disconnected mode) are displayed in the ActiveX report viewer (not the DHTML report viewer). Finally, reports can be generated in the batch mode (schedule) only in the connected mode of the Dedicated Web Client.

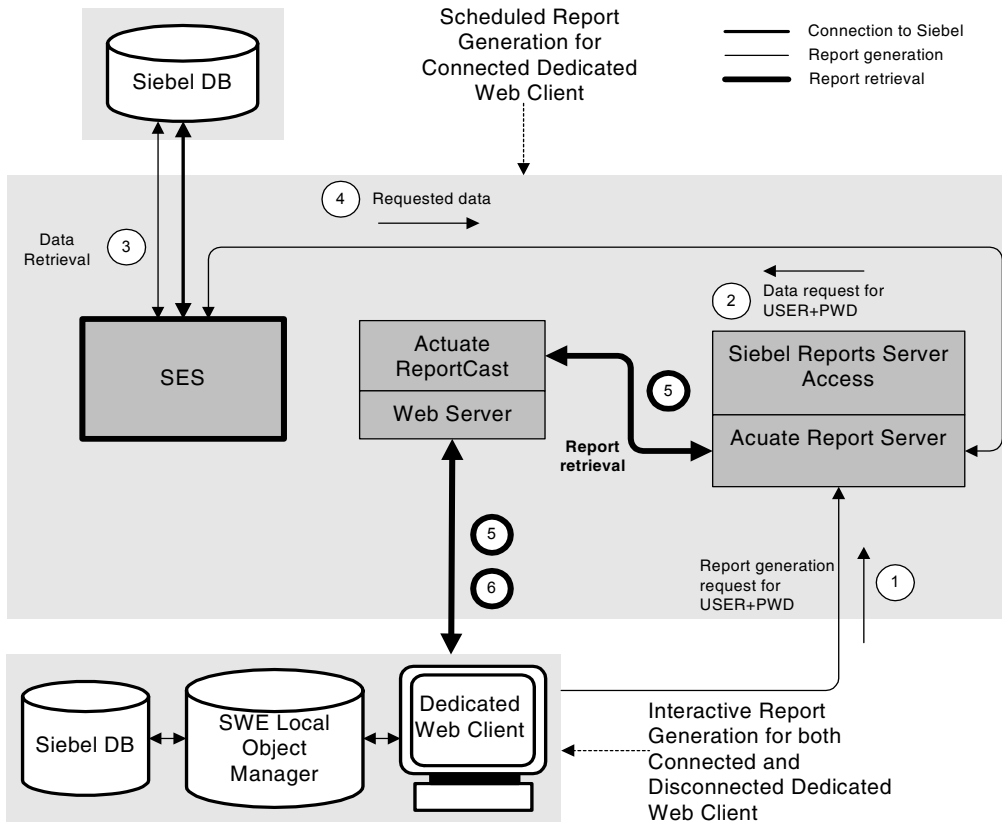
### System Architecture

In this section, the report execution in the Dedicated Web Client is described. The Reports Server encyclopedia and the Client consist of the report executables. These report executables correspond to the available reports among all the views.

When a report is run interactively in the Dedicated Web Client (connected or disconnected mode) the corresponding report executable is launched in the Client. The data for the report is obtained from the local Object Manager (see [Figure 1-5](#)), and the report is displayed in the ActiveX viewer in the Client.

In the connected mode of Dedicated Web Client, users can submit report requests to be run in batch mode in the Reports Server. These requests get submitted to the Reports Server in the form of a parameter (ROV) file. At report execution time, the report executable obtains data from the Siebel Object Manager and generates the report output file (ROI). The report output (ROI) file will be stored in the user folder in the reports encyclopedia and can be accessed from the Reports Server views in Siebel eBusiness applications.

Figure 1-5 illustrates the report execution in the Dedicated Web Client.



**Figure 1-5. Server-Based Reporting for the Siebel Dedicated Web Client**

The following is the sequence of report generation steps (corresponding to Figure 1-5) in batch mode in the connected Dedicated Web Client:

- 1 The report request is submitted to the Reports Server through the SES (Siebel Enterprise Server).
- 2 The Reports Server runs the report executable and requests data from the SES through Siebel Reports Server Access.
- 3 Data is retrieved by the Siebel Object Manager from the database and provided to the Reports Server.

- 4 The Siebel Object Manager provides data to the Reports Server. Report is generated and saved in the Reports Server encyclopedia to be retrieved by the user on demand.
- 5 The Dedicated Web Client connects to Actuate ReportCast, which authenticates the user on the Reports Server.
- 6 The report, in DHTML, is displayed in the Dedicated Web Client.

---

**NOTE:** The user need not be logged into a Siebel eBusiness application for the scheduled reports to be generated since report generation by the Reports Server is asynchronous and data for report generation is obtained from the Siebel Object Manager.

---

## User Interface and Process Flow

### Connected Mode

Dedicated Web Client reporting in the connected mode is identical to the procedures described in the “Reporting in the Web Client” on page 1-4.

### Disconnected Mode

Reports can be generated in the Dedicated Web Client in the disconnected mode also. However, the data for report generation is obtained from the local object manager. The report generation process in the disconnected mode differs from the connected mode as described below:

- Reports are run locally by obtaining data from the local database.
- Reports cannot be scheduled.
- Run Report Time Limit message is not available.

## Siebel Reports Server Views

The views described in this section appear in the Site Map (View > Site Map) under Reports Server in a Siebel eBusiness application. The only exception is the Reports Server Admin Profile view, which appears under Application Administration on the Site Map. While User Administration and Reports Server Admin Profile views are available only to the reports administrator, all other views are available to the end users.

Table 1-2 displays the Reports Server views available to the end users and administrator. An “x” symbolizes that a view is available. A blank space symbolizes a view is not available.

**Table 1-2. Availability of Reports Server Views**

View	End-User	Administrator
User Administration		x
Reports Server Admin Profile		x
Personal Profile	x	x
Scheduled Requests	x	x
Completed Requests Notifications	x	x
Active Requests	x	x
Explorer	x	x
My Reports	x	x

### User Administration View

The User Administration view is only available to an administrator. This view allows the administrator to automatically create and synchronize accounts for Siebel users in the Actuate reports encyclopedia. The administrator can synchronize one account or multiple accounts at once from this view on an ongoing basis. It is recommended that the administrator run a query in this view to synchronize users in smaller buckets particularly when large number of users should be synchronized.

As the result of synchronization, an account with the same user ID that appears in the Siebel applications is created on the Reports Server. Also, a default folder with this name is created on the Reports Server, and basic privileges such as read, write, and execute are granted to that user to access the folder. Each Reports Server account is created with the same user ID as the Siebel user ID and a password. This user ID and password are stored in an encrypted format in the Siebel database, to be passed to the Reports Server when needed.

The navigation path to this view is View > Site Map > Reports Server > User Administration. An administrator account is established in the Actuate encyclopedia as part of post-installation tasks of the Reports Server (see *Siebel Server Installation Guide for Microsoft Windows* or *Siebel Server Installation Guide for UNIX*). This account allows the administrator to log in to the Actuate Administrator Desktop to perform ongoing administration tasks such as administering the encyclopedia and cleaning up the encyclopedia. For more details please see “[Maintaining the Actuate e.Reporting Server](#)” section in *Siebel Reports Administration Guide*.

The User Administration view is shown in [Figure 1-6](#).

The screenshot shows a web application interface for Siebel User Administration. At the top, there is a navigation bar with a 'Show:' dropdown menu set to 'User Administration', a search bar, and a 'Favorites:' dropdown. Below this is a 'Reports Server:' label. The main content area is titled 'Siebel Users' and contains a table with four columns: 'Last Name', 'First Name', 'User ID', and 'Job Title'. The table has 15 rows of data, with the first row (Marino, Mille, MMARINO, Manager Sales Ops) highlighted in yellow. Above the table, there are buttons for 'Query', 'Synchronize All', and 'Synchronize One', along with a status indicator '1 - 15 of 15\*'. The table data is as follows:

Last Name	First Name	User ID	Job Title
Marino	Mille	MMARINO	Manager Sales Ops
Cain	Dave	HKIM	Manager Sales Ops
Beale	Andrea	AROBERTS	Director eBusiness Operations
Malibu	Dianna	DMALIBU	Director of Telemarketing
eSales	Consumer	BSTEVENS	Field Sales Representative
Osaka	Moiyoku	MOSAKA	Presales Representative
Hansen	Javier	DBROWN	Telemarketing Representative
Tenenholtz	Haydn	SSMITH	Telemarketing Representative
Stamey	Dave	DSTAMEY	Telemarketing Representative
Clark	Lilly	LCLARK	Swedish Admin
Wagg	Curtis	ASWAN	Field Sales Representative
Cook	Bruno	BCOOK	Field Sales Representative
Jones	Mike	MJONES	Field Sales Representative
Franks	Katherineen	KFRANKS	International Sales Representative
Towson	Ellis	ETOWNSON	Field Sales Representative

**Figure 1-6. User Administration View**

This view has the following buttons and list columns:

- **Last Name.** The last name of the user, as stored in the employee table.
- **First Name.** The first name of the user, as stored in the employee table.
- **User ID.** The Siebel user ID for each user, as stored in the employee table.
- **Job Title.** The position of the employee.
- **Menu icon.** The menu contains actions that apply directly to the applet.
- **Query button.** Click this button to generate a new form for entering the query criteria. The query results are displayed in the same applet from which the query was launched.
- **Synchronize All button.** Click this button to create accounts (referred to as synchronization) for all displayed users in the Actuate encyclopedia. If a large number of users should be synchronized, please run a query to restrict the displayed list of users and then click Synchronize All.
- **Synchronize One button.** Click this button to transfer the currently selected user ID to Actuate encyclopedia.

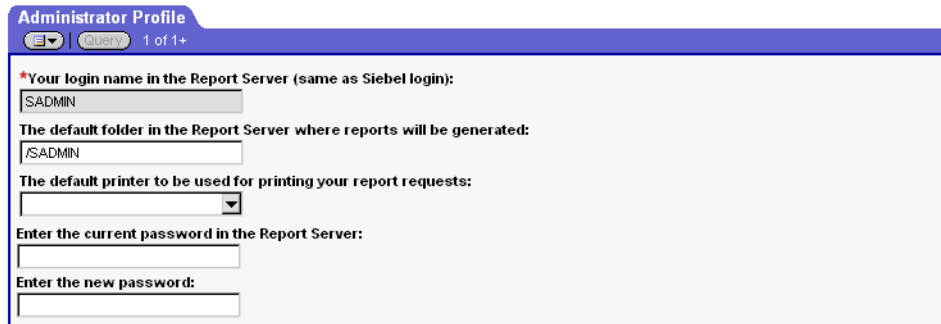
## Reports Server Admin Profile View

The Reports Server Admin Profile view is also only available to an administrator. This view allows an administrator to establish and change their password, to specify a default printer, and to select the directory in which the report output files are saved.

This view, unlike the others, is not under Reports Server on the Site Map (View > Site Map). This view is under Application Administration on the Site Map.

The Reports Server Admin Profile view is shown in [Figure 1-7](#).

Reports Server:



Administrator Profile

Query 1 of 1+

\*Your login name in the Report Server (same as Siebel login):  
SADMIN

The default folder in the Report Server where reports will be generated:  
/SADMIN

The default printer to be used for printing your report requests:  
▼

Enter the current password in the Report Server:  
\_\_\_\_\_

Enter the new password:  
\_\_\_\_\_

**Figure 1-7. Reports Server Admin Profile View**

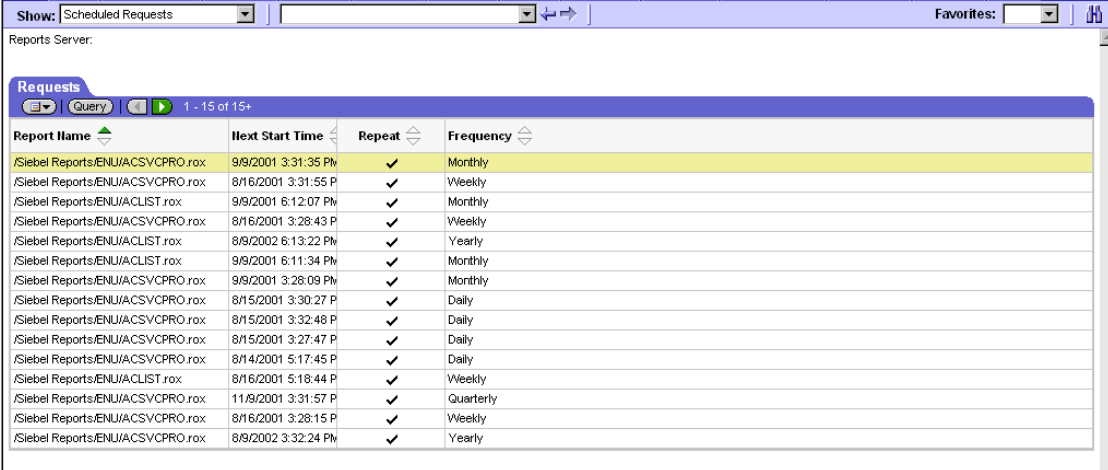
This view contains the following fields:

- **Menu icon.** The menu contains actions that apply directly to the applet.
- **Query button.** Not used on this view.
- **Login Name.** This field displays the login name of the currently logged in user. It is read-only.
- **Default folder.** The administrator's default Report Encyclopedia folder for storage of report output. By default, this folder name is the same as the administrator account. The administrator can specify an existing folder name other than the default folder name.
- **Default printer.** The administrator may select the default printer to display in the schedule dialog.
- **Current Password.** Current password of the administrator.
- **New Password.** If an administrator wishes to change their password, the new password should be entered in this field.

## Scheduled Requests View

The Scheduled Requests view displays the report requests scheduled by the user. If a report is frequently run, the next start time and frequency for that report is displayed in this view.

The Scheduled Requests view is shown in [Figure 1-8](#).



Report Name	Next Start Time	Repeat	Frequency
/Siebel Reports/ENU/ACSVCPRO.rox	9/9/2001 3:31:35 PM	<input checked="" type="checkbox"/>	Monthly
/Siebel Reports/ENU/ACSVCPRO.rox	8/16/2001 3:31:55 P	<input checked="" type="checkbox"/>	Weekly
/Siebel Reports/ENU/ACLIST.rox	9/9/2001 6:12:07 PM	<input checked="" type="checkbox"/>	Monthly
/Siebel Reports/ENU/ACSVCPRO.rox	8/16/2001 3:28:43 P	<input checked="" type="checkbox"/>	Weekly
/Siebel Reports/ENU/ACLIST.rox	8/9/2002 6:13:22 PM	<input checked="" type="checkbox"/>	Yearly
/Siebel Reports/ENU/ACLIST.rox	9/9/2001 6:11:34 PM	<input checked="" type="checkbox"/>	Monthly
/Siebel Reports/ENU/ACSVCPRO.rox	9/9/2001 3:28:09 PM	<input checked="" type="checkbox"/>	Monthly
/Siebel Reports/ENU/ACSVCPRO.rox	8/15/2001 3:30:27 P	<input checked="" type="checkbox"/>	Daily
/Siebel Reports/ENU/ACSVCPRO.rox	8/15/2001 3:32:48 P	<input checked="" type="checkbox"/>	Daily
/Siebel Reports/ENU/ACSVCPRO.rox	8/15/2001 3:27:47 P	<input checked="" type="checkbox"/>	Daily
/Siebel Reports/ENU/ACSVCPRO.rox	8/14/2001 5:17:45 P	<input checked="" type="checkbox"/>	Daily
/Siebel Reports/ENU/ACLIST.rox	8/16/2001 5:18:44 P	<input checked="" type="checkbox"/>	Weekly
/Siebel Reports/ENU/ACSVCPRO.rox	11/9/2001 3:31:57 P	<input checked="" type="checkbox"/>	Quarterly
/Siebel Reports/ENU/ACSVCPRO.rox	8/16/2001 3:28:15 P	<input checked="" type="checkbox"/>	Weekly
/Siebel Reports/ENU/ACSVCPRO.rox	8/9/2002 3:32:24 PM	<input checked="" type="checkbox"/>	Yearly

**Figure 1-8. Scheduled Requests View**

This view contains the following buttons and list columns:

- **Menu icon.** The menu contains actions that apply directly to the applet.
- **Query button.** Click this button to run a query on the displayed records. The query results are displayed in the same applet from which the query was launched.
- **Report Name.** Name of the report executable to be executed.
- **Next Start Time.** Date and time that the scheduled report will start running.
- **Repeat.** Checked for a periodic report, and unchecked for all one-time scheduled reports.
- **Frequency.** Not applicable if a one-time report is scheduled (the field is left blank). The frequency (Daily, Weekly and so on) of the scheduled report is indicated in this field.

## Completed Request Notifications View

The Completed Request Notifications view provides information about the status of completed requests including the status of that report. All completed requests include a hyperlink to the report output.

The Completed Request Notifications view is shown in [Figure 1-9](#).

Report Name	Version	Pages	Status	Completed	Status Detail
	0	0	Failure	8/14/2001 3:33:08 P	Unable to get system default printer name from Administration Server.
	0	0	Failure	8/14/2001 3:28:10 P	Unable to get system default printer name from Administration Server.
/SADMIN/ACSYCPR	12	362	Success	8/14/2001 1:22:45 P	
/SADMIN/ORSUMAL	1	3	Success	8/14/2001 10:48:57	
/SADMIN/EMPSCHET	4	79	Success	8/13/2001 3:50:10 P	
/SADMIN/EMPSCHET	3	79	Success	8/13/2001 3:43:12 P	
	0	0	Failure	8/13/2001 3:33:05 P	Unable to get system default printer name from Administration Server.
/SADMIN/EMPSCHET	2	79	Success	8/13/2001 3:32:47 P	
	0	0	Failure	8/13/2001 3:28:05 P	Unable to get system default printer name from Administration Server.
/SADMIN/EMPSCHET	1	79	Success	8/13/2001 3:27:39 P	
/SADMIN/print.roi	1	2	Success	8/13/2001 12:19:25	
/SADMIN/PROJSTAT	7	2	Success	8/13/2001 12:09:09	
/SADMIN/PROJSTAT	6	2	Success	8/13/2001 12:01:46	
/SADMIN/PROJSTAT	5	508	Success	8/13/2001 11:42:26	
/SADMIN/CNTLIST.rc	2	233	Success	8/13/2001 11:03:49	

**Figure 1-9. Completed Request Notifications View**

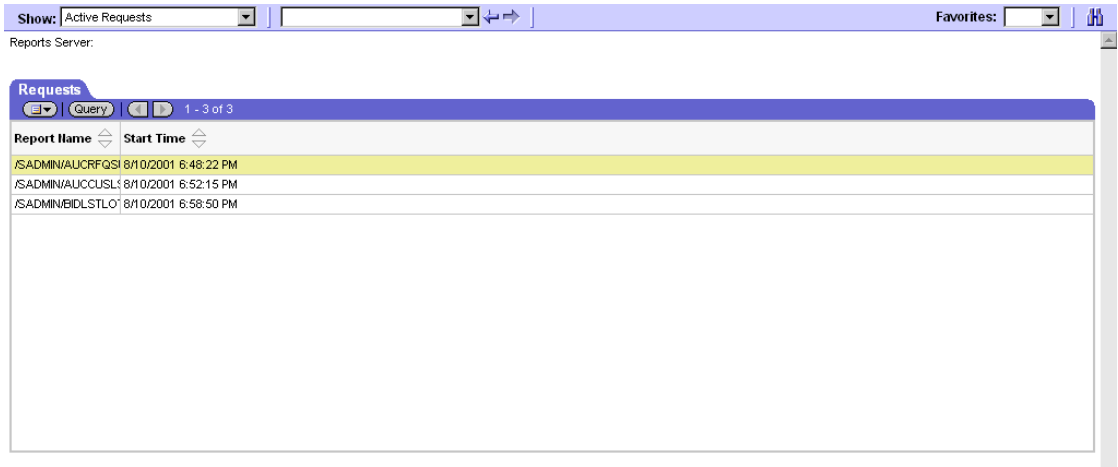
This view has the following buttons and list columns:

- **Menu icon.** The menu contains actions that apply directly to the applet.
- **Query button.** Click this button to run a query on the displayed records in the view. The query results are displayed in the same applet from which the query was launched.
- **Report Name.** If the request was successful, a link to the report output (ROI) file appears here. Clicking the hyperlink displays the report in the DHTML Report viewer. If the report generation was unsuccessful, the report name field is left blank.
- **Version.** Version number of this report output. Each time a report is run, a different version of that report is created. The highest version is the most recently run report.
- **Pages.** Number of pages in the generated report, if it was successful.
- **Status.** Success or Failure, depending on the outcome of the request.
- **Completed.** Date and time that the report completed or aborted.
- **Status Detail.** Explanation of the reason for failure, if this is an unsuccessful report run.

### Active Requests View

The Active Requests view provides a list of reports that are currently being generated. This view provides a means to check the status of large reports.

The Active Requests view is shown in [Figure 1-10](#).



**Figure 1-10. Active Requests View**

This view contains the following buttons and list columns:

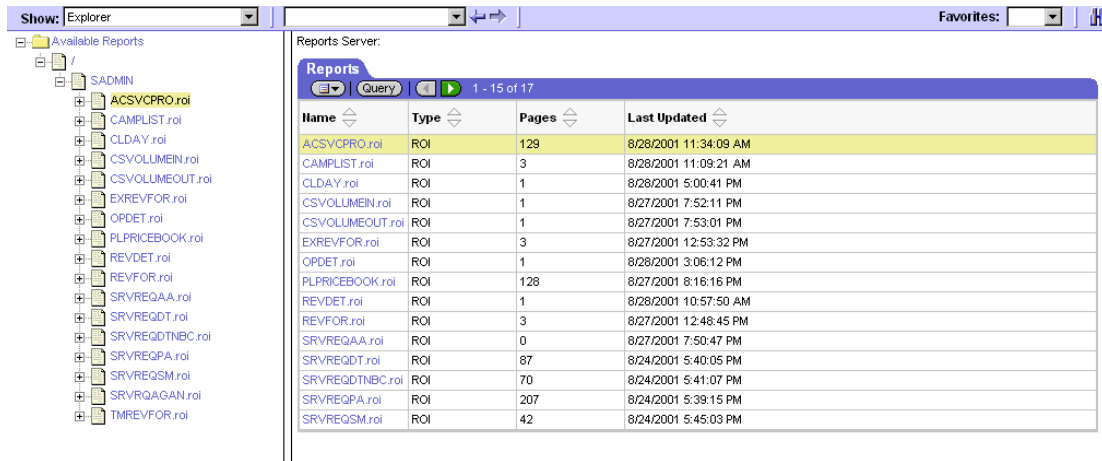
- **Menu icon.** The menu contains actions that apply directly to the applet.
- **Query button.** Click this button to run a query on the records displayed in this view. The query results are displayed in the same applet from which the query was launched.
- **Report Name.** Name of the report executable that is being run.
- **Start Time.** Date and time when report generation began.

## Explorer View

The Explorer view displays a tree applet displaying the contents in the user's folder, and all other accessible reports in the encyclopedia, and a list applet displaying the related contents. Only those reports to which the user has access are shown.

In the tree applet (on the left-hand side) of the Explorer view, the reports server folders are displayed hierarchically. The top-level (root) folder is identified with a slash (/) symbol, and its top-level subfolders correspond to user account names. The top-level folders may have administrator-defined subfolders. When a folder in the tree applet is selected, the list applet on the right hand side displays reports in that folder.

The Explorer view is shown in [Figure 1-11](#).



**Figure 1-11. Explorer View**

This view contains the following buttons and list columns:

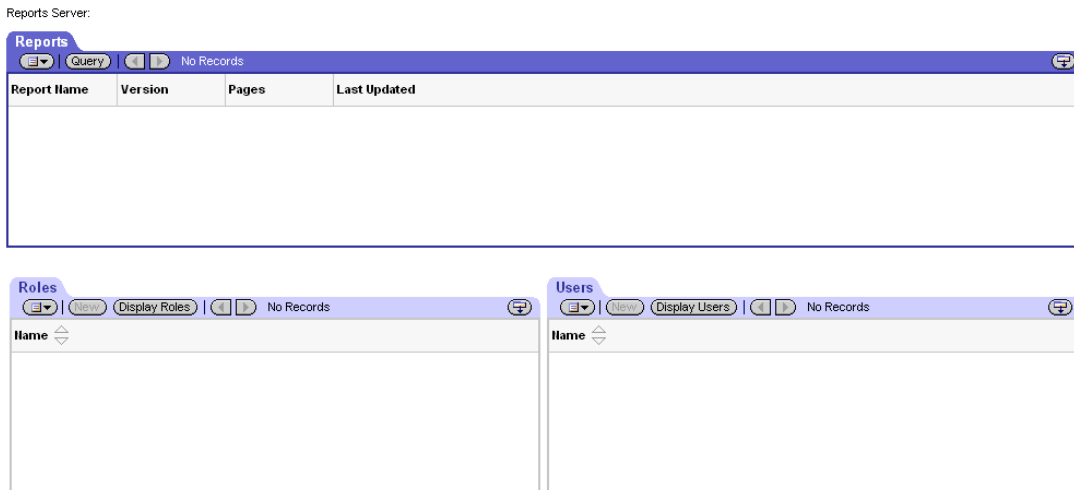
- **Menu icon.** The menu contains actions that apply directly to the applet.
- **Query button.** Click this button to run a query on the displayed records in the view. The query results are displayed in the same applet from which the query was launched.
- **Name.** Either a subfolder name or a hyperlink to the report output (ROI) file. Clicking on the report name hyperlink displays the report in a DHTML report viewer.
- **Type.** Indicates the type of contents displayed in the list applet. If reports are displayed, the Type field displays a ROI file extension. If a folder is displayed, “folder” is displayed in the Type field.
- **Pages.** Number of pages in the generated report.
- **Last Updated.** Date and time of the report run that generated this version.

## My Reports View

The My Reports view displays the report output files that the user has access to in the Reports applet. This includes the reports generated by the user and the reports that other users granted permission to view.

The view also displays Users and Roles applets at the bottom of the view. The user may grant permission to other users and or roles to view and print a report using these applets. First the user selects a report from the Reports applet. Then, users and roles are added by creating new records in the Users and Roles applets, respectively.

The My Reports view is shown in [Figure 1-12](#).



**Figure 1-12. My Reports View**

This view contains the following buttons and list columns:

- **Menu icon.** The menu contains actions that apply directly to the applet.
- **Query button.** Click this button to run a query on the displayed records in the view. The query results are displayed in the same applet from which the query was launched.
- **Report Name.** The full pathname to the report output (ROI) file appearing as a hyperlink. Clicking the hyperlink displays the report in a DHTML report viewer.
- **Version Number.** Displays the version number of the report if the same report was generated multiple number of times. The highest-numbered version is the most recent.
- **Pages.** Number of pages in the generated report.
- **Last Updated.** Date and time of the report run that generated this version.
- **Display Roles button.** Displays a list of roles that have been granted access to view a specific report. Located on the Roles applet.

- **Display Users button.** Displays a list of users that have been granted access to view a specific report. Located on the Users applet.

## Personal Profile View

The Personal Profile View allows a user to specify a default printer and file system folder for report output.

The Personal Profile view is shown in [Figure 1-13](#).

The screenshot shows a web application interface. At the top, there is a navigation bar with a 'Show:' dropdown menu set to 'Personal Profile', a search input field, and a 'Fav' button. Below this, the text 'Reports Server:' is visible. The main content area is titled 'Personal Profile' and contains a 'Save' button and '1 of 1' indicator. The form fields are:

- Your login name in the Report Server (same as Siebel login):** A text input field containing 'SADMIN'.
- The default folder in the Report Server where reports will be generated:** A text input field containing '/SADMIN'.
- The default printer to be used for printing your report requests:** A dropdown menu.

**Figure 1-13. Personal Profile View**

The Personal Profile form contains the following fields:

- **Menu icon.** The menu contains actions that apply directly to the applet.
- **Save button.** After entering changes, click Save to save the information.
- **Login Name.** This field displays the login name of the currently logged in user. It is read-only.
- **Default folder.** The user's default folder on the report encyclopedia for storing report output files. By default, this folder name is the same name as the user ID. The user may specify a folder name other than the default folder, but it must already exist on the report encyclopedia, and the user must have access to it.
- **Default printer.** The printer to which the report will be sent if the user selected print option in the schedule dialog while scheduling a report. This printer name appears in the schedule dialog when the user is scheduling a report. The user can choose a different printer at that time.

The following chapter was not included in Siebel Reports Administration Guide:

## Restructuring the Reports Server Encyclopedia

### About This Chapter

This chapter describes the process a Siebel Reports Server administrator must follow to restructure the Reports Server Encyclopedia.

As a result of the synchronization process, folders for reports users are created in the Reports Server encyclopedia at the root level. Therefore, synchronizing 5,000 reports users will result in creating 5,000 folders at the root level of the encyclopedia. Actuate strongly recommends that folders at the root level of the encyclopedia should be minimized to ensure better response time and performance of the Reports Server. Therefore, Siebel created a utility to automatically restructure the encyclopedia such that only a small number of folders exist at the root level. Each of the root level folders contains a certain number of user folders as chosen by the administrator. The following describes the installation and usage of this utility.

### Installation of the `actrestruct5.exe` Utility

The following steps describe the installation of the `actrestruct5.exe` utility:

#### **To install the `actrestruct5.exe` utility**

- 1 Insert the Windows Client Software CD-ROM into the CD-ROM drive of the Windows machine in which Actuate Administrator Desktop was installed.
- 2 Navigate to the following directory:

```
seatools_and_windows_client_ancillary\Thirdpty\language\actuate\actutil
```

Where *language* = the code for the language pack you installed; *enu* for all languages except Japanese, which is *jpn*.

- 3 Copy `actrestruct5.exe` into the `\bin` directory of the Actuate Administrator Desktop folder, for example `C:\Actuate5\adt\bin\`.

### Usage of the `actrestruct5.exe` Utility

The following describes the usage of the `actrestruct5.exe` utility.

The administrator should run the utility `actrestruct5.exe` from a Windows DOS prompt on a machine where Actuate Administrator Desktop is installed. The syntax for running this utility is:

```
actrestruct.exe -u <user name> -p <password> -m <machine> - n <starting folder>
-r <root level folders> -s <subfolders>
```

Flag	Type	Description
-u	Required	Reports Server administrator login name.
-p	Required	Reports Server administrator password.
-m	Required	Reports Server host name.
-n	Optional	Integer specifying the root level folder to start with for creating subfolders. If omitted, subfolders will be created starting with the first root level user folder.  It is important to specify an appropriate value for this parameter if this utility is run periodically.
-r	Optional	Number of root level folders (of the form foldernnn) that will be created. Existing folders will not be affected. Default value is 50.
-s	Optional	Number of subfolders per root-level folder. Default value is 250.

The following represent sample usage scenarios.

**Example 1:** The administrator has synchronized 8,000 users on the Reports Server and then run the `actrestruct.exe` utility as shown below:

```
actrestruct5.exe -u administrator -p admin -m Server1 -n 1 -r 40 -s 400
```

**Result:**

There will be 40 folders created at the root level in the Server1 Reports Server encyclopedia of the form folder001, folder002, ..., folder040. Each of these root level folders contains the folders of 400 users. The first 400 user folders are created in folder001; the next 400 user folders are created in folder002 and so on. The root level folders are created first, so all 40 folders will be created at the root level. However only 20 of them will be used to create subfolders. The remaining root level folders can be used to populate subfolders when this utility is run subsequent times.

**Example 2:** A couple of weeks later, 500 new employees joined the company. As usual, the administrator synchronized these 500 users on the Reports Server. Then, the administrator ran the `actrestruct.exe` utility using the following command.

```
actrestruct5.exe -u administrator -p admin -m Server1 -n 21 -s 400
```

**Result:**

At the root level in the Reports Server encyclopedia of Server1, folder021 and folder022 are used to create subfolders for the new users. The folder021 contains the first 400 user folders, and folder022 contains the remaining 100 user folders.

---

**NOTE:** In Example 2, if the administrator specifies -n 20, the 400 new user folders will be added to folder020, which will then contain 800 folders. The folder021 will contain the remaining 100 user folders.

---

# Chapter 1, Getting Started

December 28, 2001

## Search Specification and Sort Specification properties

*On page 1-19, add the following to Search Specification property section:*

Also, if you set a search specification on your parent datastream in Actuate Start(), this search spec will override the dynamic view query results in the report and look through all the records in the business component.

*On page 1-25, add this information to the note following the section on static reports:*

---

**NOTE:** Also, a parent business component for a report will still use a sort specification property even though it has an asterisk that says for Static reports.

---

## Printing

*On page 1-28, add the following information:*

If you set the paper size in the report design, the setting is ignored when the report is printed. The report will always use the default setting in the print setup. Also, the setting in the report design can only take effect if the Actuate API. However, the use of this is not recommended in Siebel eBusiness applications.

For more information regarding printing, see Actuate's *Using e.Reports* manual under the "Third Party" section of the *Siebel Bookshelf, MidMarket Edition* CD-ROM.

# Appendix B, Method Reference

**December 28, 2001**

*For Actuate 5, the Start, Fetch, and Finish methods are changed. Replace the code samples as stated.*

*Replace the Start Method on page B-3 with the following code sample:*

```
Function Start () As Boolean

    Dim bCurRowOnly As Boolean
    Dim bReExecute As Boolean
    Dim bUpdateLinks As Boolean
    Dim bStatus As Boolean
    Dim mainRowId As String
    Dim busObjName As String
    Dim SortSpecDyn As String
    Dim searchSpec As String
    Dim curViewMode As Integer

    bCurRowOnly = False
    baseReport::bCurRowOnly = bCurRowOnly
    bReExecute = False
    baseReport::bQueryExecuted = False
    bUpdateLinks = False
    errCode = 0
    mainRowId = ""
    busObjName = ""

    bStatus = Super::Start ()
    If (bStatus = False) Then
        Start = False
        Exit Function
    End If

    ssAppServer = ssReport::ssiSiebelServer

    ssModelSetPositionId (ssAppServer, ssReport::ssPositionId)

    ssBO = ssModelGetBusObject (ssAppServer, ssReport::ssBusObjectName)
    ssReport::ssActiveBusObject = ssBO

    ' SearchSpec is recieved depending on where it is originated
    SearchSpec = ""
    If (ssReport::ssSearchSpec <> "") And (errCode = 0) Then searchSpec =
ssReport::ssSearchSpec

    ' SortSpec is recieved depending on where it is originated
    SortSpecDyn = "Name"
```

```

If (ssReport::ssSortSpec <> "") And (errCode = 0) Then SortSpecDyn =
ssReport::ssSortSpec

If (errCode = 0) Then ssAccount = ssBusObjGetBusComp (ssBO, "Account")
If (errCode = 0) Then ssBusCompSuppressNotification (ssAccount)
If (errCode = 0) And (ssType = 2) Then ssBusCompDeactivateFields (ssAccount)
If (errCode = 0) Then curViewMode = ssBusCompGetViewMode (ssAccount)

' If the main report BusComp has a search or sort spec,
' or uses a field that isn't already active, its query
' must be re-executed.
If (ssType = 2) Then bReExecute = True
If (Not bReExecute) Then bReExecute = (searchSpec <> "") Or (SortSpecDyn <> "")
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "IsExecuted", "") = "N")
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "IsFieldActive", "Account Status") = "N")
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "IsFieldActive", "City") = "N")
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "IsFieldActive", "Competitor") = "N")
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "IsFieldActive", "Country") = "N")
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "IsFieldActive", "Description") = "N")
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "IsFieldActive", "Industry") = "N")
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "IsFieldActive", "Location") = "N")
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "IsFieldActive", "Main Fax Number") = "N")
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "IsFieldActive", "Main Phone Number") = "N")
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "IsFieldActive", "Name") = "N")
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "IsFieldActive", "Parent Account Name") = "N")
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "IsFieldActive", "Postal Code") = "N")
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "IsFieldActive", "State") = "N")
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "IsFieldActive", "Street Address") = "N")
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "IsFieldActive", "Synonym") = "N")
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "IsFieldActive", "Type") = "N")
' Last check if business component needs to be re-executed
If (errCode = 0) And (Not bReExecute) Then bReExecute = (ssBusCompInvokeMethod
(ssAccount, "ReExecuteReport", "") = "Y")
' Cache the current row-id so it can be re-queried

```

```

If (bCurRowOnly) Then mainRowId = ssReport::ssActiveRowId

If (errCode = 0) And (ssType = 1) And (bReExecute) Then
' To re-execute a current-row-only report, the active
' BusComp need not be disturbed. Another BusObj can
' be constructed just for this query.
' This is NOT applicable for server based report generation.
If (bCurRowOnly) And (ssReport::ssBusObjectName = "") Then
    baseReport::bQueryExecuted = True
    If Not ( ssAccount = 0 ) Then ssBusCompAllowNotification (ssAccount)
    If (errCode = 0) Then busObjName = ssBusObjGetName (ssBO)
    If (errCode = 0) Then ssBO = ssModelGetBusObject (ssAppServer, busObjName)
    If (errCode = 0) Then ssAccount = ssBusObjGetBusComp (ssBO, "Account")
    If (errCode = 0) Then ssBusCompSetViewMode (ssAccount, curViewMode)
    If (errCode = 0) Then ssBusCompSuppressNotification (ssAccount)
End If
End If

' Now actually setup the main report BusComp
If (errCode = 0) Then ssBusCompActivateField (ssAccount, "Account Status")
If (errCode = 0) Then ssBusCompActivateField (ssAccount, "City")
If (errCode = 0) Then ssBusCompActivateField (ssAccount, "Competitor")
If (errCode = 0) Then ssBusCompActivateField (ssAccount, "Country")
If (errCode = 0) Then ssBusCompActivateField (ssAccount, "Description")
If (errCode = 0) Then ssBusCompActivateField (ssAccount, "Industry")
If (errCode = 0) Then ssBusCompActivateField (ssAccount, "Location")
If (errCode = 0) Then ssBusCompActivateField (ssAccount, "Main Fax Number")
If (errCode = 0) Then ssBusCompActivateField (ssAccount, "Main Phone Number")
If (errCode = 0) Then ssBusCompActivateField (ssAccount, "Name")
If (errCode = 0) Then ssBusCompActivateField (ssAccount, "Parent Account Name")
If (errCode = 0) Then ssBusCompActivateField (ssAccount, "Postal Code")
If (errCode = 0) Then ssBusCompActivateField (ssAccount, "State")
If (errCode = 0) Then ssBusCompActivateField (ssAccount, "Street Address")
If (errCode = 0) Then ssBusCompActivateField (ssAccount, "Synonym")
If (errCode = 0) Then ssBusCompActivateField (ssAccount, "Type")

If (errCode = 0) Then ssBusCompInvokeMethod (ssAccount,
"SetIgnoreMaxCursorSize", "Y")

If (errCode = 0) And (ssType = 2) And (bCurRowOnly = FALSE) Then
    If (baseReport::bSetForward) Then
        ssBusCompInvokeMethod(ssAccount, "GotoBookmarkT", ssReport::ssBookmark)
    Else
        ssBusCompInvokeMethod(ssAccount, "GotoBookmarkF", ssReport::ssBookmark)
    End If
End If

If (errCode = 0) And (searchSpec <> "") And ((Not bCurRowOnly) Or
(ssReport::ssBusObjectName <> "")) Then
    ssBusCompSetSearchExpr (ssAccount, searchSpec)

```

```

End If

If (errCode = 0) And (SortSpecDyn <> "") Then
    ssBusCompSetSortSpec (ssAccount, SortSpecDyn)
End If
If (errCode = 0) And (ssReport::ssBusObjectName <> "") And (ssReport::ssViewMode
<> "") Then ssBusCompSetViewMode (ssAccount, CInt (ssReport::ssViewMode))

' Re-execute business components as necessary
If (errCode = 0) Then
    If (bReExecute) Then
        If (bCurRowOnly) And (mainRowId <> "") Then
            ssBusCompSetSearchExpr
(ssAccount, "Id = "" + mainRowId + """)
            If (errCode = 0) Then ssBusCompExecuteQuery2 (ssAccount,
baseReport::bSetForward, True)
            ElseIf (bUpdateLinks) Then
                ssBusCompInvokeMethod (ssAccount, "UpdateLinks", "")
            End If
        End If
    End If

' Process errors and return
If (errCode = 0) Then
    bStatus = True
Else
    bStatus = False
    ssProcessLastError(ssAppServer, "", "")
End If

Start = bStatus
End Function

```

Replace the Fetch method on page B-8 with the following code sample:

```

Function Fetch () As AcDataRow

    Dim    bStatus      As Boolean
    Dim    bCurRowOnly As Boolean
    Dim    custDataRow  As ssAccountDataRow
    Dim    theBC        As Integer

    errCode = 0
    bStatus = False
    bCurRowOnly = False
    theBC = ssAccount

    If (bCurRowOnly = True) Then
        If (Position = 1) Then
            If (baseReport::bQueryExecuted = True) Then
                bStatus = ssBusCompFirstRecord (theBC)
            Else

```

```

        bStatus = True
    End If
End If
Else
    If (Position = 1) Then
        bStatus = ssBusCompFirstRecord (theBC)
        If (errCode = 0) And (baseReport::bSetForward) Then ssBusCompInvokeMethod
(theBC, "SetForwardOnly", "")
    Else
        bStatus = ssBusCompNextRecord (theBC)
    End If
End If

If (bStatus = True) And (errCode = 0) Then
    Set custDataRow = NewDataRow

    If (errCode = 0) Then custDataRow.ssAccount_Status = ssBusCompGetFieldValue
(theBC, "Account Status")
    If (errCode = 0) Then custDataRow.ssCity = ssBusCompGetFieldValue (theBC,
"City")
    If (errCode = 0) Then custDataRow.ssCompetitor = ssBusCompGetFieldValue
(theBC, "Competitor")
    If (errCode = 0) Then custDataRow.ssCountry = ssBusCompGetFieldValue (theBC,
"Country")
    If (errCode = 0) Then custDataRow.ssDescription = ssBusCompGetFieldValue
(theBC, "Description")
    If (errCode = 0) Then custDataRow.ssIndustry = ssBusCompGetFieldValue (theBC,
"Industry")
    If (errCode = 0) Then custDataRow.ssLocation = ssBusCompGetFieldValue (theBC,
"Location")
    If (errCode = 0) Then custDataRow.ssMain_Fax_Number =
ssBusCompGetFormattedFieldValue (theBC, "Main Fax Number")
    If (errCode = 0) Then custDataRow.ssMain_Phone_Number =
ssBusCompGetFormattedFieldValue (theBC, "Main Phone Number")
    If (errCode = 0) Then custDataRow.ssName = ssBusCompGetFieldValue (theBC,
"Name")
    If (errCode = 0) Then custDataRow.ssParent_Account_Name =
ssBusCompGetFieldValue (theBC, "Parent Account Name")
    If (errCode = 0) Then custDataRow.ssPostal_Code = ssBusCompGetFieldValue
(theBC, "Postal Code")
    If (errCode = 0) Then custDataRow.ssState = ssBusCompGetFieldValue (theBC,
"State")
    If (errCode = 0) Then custDataRow.ssStreet_Address = ssBusCompGetFieldValue
(theBC, "Street Address")
    If (errCode = 0) Then custDataRow.ssSynonym = ssBusCompGetFieldValue (theBC,
"Synonym")
    If (errCode = 0) Then custDataRow.ssType = ssBusCompGetFieldValue (theBC,
"Type")

    ' Now retrieve the system fields
    If (errCode = 0) Then custDataRow.ssId = ssBusCompGetFieldValue (theBC, "Id")

```

```

        If (errCode = 0) Then custDataRow.ssCreated = ssBusCompGetFieldValue (theBC,
"Created")
        If (errCode = 0) Then custDataRow.ssCreated_Formatted =
ssBusCompGetFormattedFieldValue (theBC, "Created")
        If (errCode = 0) Then custDataRow.ssCreated_By = ssBusCompGetFieldValue
(theBC, "Created By")
        If (errCode = 0) Then custDataRow.ssUpdated = ssBusCompGetFieldValue (theBC,
"Updated")
        If (errCode = 0) Then custDataRow.ssUpdated_Formatted =
ssBusCompGetFormattedFieldValue (theBC, "Updated")
        If (errCode = 0) Then custDataRow.ssUpdated_By = ssBusCompGetFieldValue
(theBC, "Updated By")

        Set Fetch = custDataRow
        AddRow (Fetch)
    Else
        Set Fetch = Nothing
    End If

    If (errCode <> 0) Then
        ssProcessLastError(ssAppServer, "", "")
    End If

End Function

```

Replace the Delete method on page B-11 with the following code sample:

```

Sub Delete ()

    ssRestoreActiveRow (ssAccount)
    If Not (ssAccount = 0) Then ssBusCompInvokeMethod (ssAccount,
"SetIgnoreMaxCursorSize", "N")
    If Not (ssAccount = 0) Then ssBusCompAllowNotification (ssAccount)
    ssAccount = 0

    Super::Delete ()
End Sub

```