Sage SalesLogix

Upgrading Sage SalesLogix from Version 7.5.x to 8.0

Version 8.0

Developed by Sage SalesLogix User Assistance

sage

Upgrading Sage SalesLogix from Version 7.5.x to 8.0

Address

Sage
8800 North Gainey Center Drive, Suite 200
Scottsdale, AZ 85258
USA

Version

Version 8.0

Technical Support

Technical Support is available to customers with support contracts directly from Sage Software, and to Certified Business Partners. Calls are answered during business hours, Monday through Friday, excluding holidays. Current contact information is available on the Sage SalesLogix Web site: http://www.sagesaleslogix.com

Partners with a valid technical support contract and a Web access code can request technical support electronically on the Sage SalesLogix Partner Support Portal Web site: partners.sagenorthamerica.com.

Customers with a valid technical support contract and a Web access code can request technical support electronically on the Sage SalesLogix Customer Support Portal Web site: customers.sagenorthamerica.com.

Revisions to this book are available through both Portal Web sites. Check regularly for current Sage SalesLogix product documentation.

Contents

This document provides instructions for upgrading existing Network and Web components from version 7.5.x to version 8.0. To install 8.0 or to add functionality to your current installation, refer to the *Sage SalesLogix Implementation Guide*.

Version 8.0 will uninstall any earlier versions of 7.5 before installing version 8.0.

- If you are upgrading from Sage SalesLogix version 7.5.0 or later, follow the steps in this guide.
- If you have never installed Sage SalesLogix, do not use this guide. Refer to the instructions in the *Sage SalesLogix Implementation Guide* document provided with this release.
- If you are upgrading from a version prior to version 7.5, you must first upgrade to version 7.5.0 before upgrading to version 8.0. Use the appropriate upgrade documents on the Sage SalesLogix Support Portal Web site (http://customers.sagenorthamerica.com or http://partners.sagenorthamerica.com) to upgrade to version 7.5.0.

> ⚠ DO NOT install this release on any Sage SalesLogix version earlier than 7.5.0.

Prerequisites

The following table shows software prerequisites that will automatically be installed on your computer if they are not already there. Installing these prerequisites may require your computer to restart. For additional prerequisites, see the Compatibility Checklist.

Note Performing upgrade installations using the install scripts do not install the required prerequisites. Please manually install prerequisites on each machine to be upgraded prior to running the install scripts.

Prerequisites	Admin Tools and Servers	LAN Client	Remote Office	Remote Client	Offline Web Client	Web Host	Web reporting
Windows Installer 3.1		X				X	X
Windows Installer 4.5	X		X	X	X		
Microsoft Exception Message Box	X		X	X			
Microsoft .NET Framework 2.0[1]							X
Microsoft .NET Framework 4.0[2]	X		X	X	X	X	
Microsoft .NET Framework Multi-Targeting Pack	X		X	X	X	X	
SQL Server 2005 Backward Compatibility	X		X	X	X	X	

Prerequisites	Admin Tools and Servers	LAN Client	Remote Office	Remote Client	Offline Web Client	Web Host	Web reporting
Microsoft SQL Server 2008 R2 Express RTM (Optional)	X		X	X	X		
PowerShell 1.0 for Windows Server 2003 (x86)	X		X	X	X		
Microsoft Windows Imaging Component	X		X	X	X	X	
Microsoft Visual C++ 2010 x86 Redistributable	X	X			X	X	X

Use the following workplan to guide your upgrade. Follow the tasks in order. If a task does not apply, disregard it and move on to the next. The "Required for" column indicates you must complete the task for:

- **All** - all Sage SalesLogix installations.
- **Network** - installations that include Network components.
- **Web** - installations that include Web components.
- **Remotes** - installations that include Remote users or Remote Offices.
- **Intellisync** - installations that include Intellisync for SalesLogix.
- **Customized** - installations that have customized their Sage SalesLogix environment.
- **Oracle** - installations running Sage SalesLogix on an Oracle database.
- **Microsoft SQL Server** - installations running Sage SalesLogix on a Microsoft SQL Server database.

Required for	Task Description
Prepare to Upgrade	

All	**1** Read all chapters in this document before proceeding. Understanding the content of the release and planning your upgrade are critical.
	 • Chapter 1, "Upgrade Workplan" - contains upgrade tasks explaining how to:
	  • Save your existing customizations.
	  • Upgrade all Sage SalesLogix components.
	  • Reapply customizations to your upgraded installation.
	  • Send changes to all users by releasing plugins, synchronizing changes, and/or deploying portals.
	  • Integrate any new features into your environment.
	 • Chapter 2, "Changes in this Release" - details the new features, functionality changes, and enhancements in this release. Any change that impacts customizations is identified as a [Breaking Change].
	 • Chapter 3, "Plugin Changes in this Release" - details the schema and plugin changes in this release.
	
	 Before upgrading, ensure that:
	 • Any Web portals included in your installation are version 7.5 or later and have deployed successfully.
	 • Your current Sage SalesLogix version is 7.5 or later.
	 • You review the *Compatibility Checklist* to understand the supported platforms, applications, and system requirements for this release.
	 The *Compatibility Checklist* is available on the Sage SalesLogix Support Portal Web site at http://customers.sagenorthamerica.com for customers and partners.sagenorthamerica.com for partners.

This task is required if ERPLink is installed.	**2** Verify the Sage SalesLogix installation path is correct.
	
	 If your installation includes ERPLink, the default SalesLogix installation location may have been changed. Before installing the upgrade, be sure to verify the default location to ensure files are being copied to the correct location.
	
	 To verify:
	 1 On your desktop, click **Start** > **Run**.
	 2 In the **Run** window, type **regedit** and click **OK**.
	 3 In the **Registry Editor** window, expand **HKEY_LOCAL_MACHINE** > **SOFTWARE** > **WOW6432Node**.
	 4 Select **SalesLogix**.
	 5 Verify the **Path** key's value:
	  • 64 bit operating systems - **C:\Program Files (x86)\SalesLogix**
	  • 32 bit operating systems - **C:\Program Files\SalesLogix**
	 6 If necessary, update the path:
	  a Right-click the **Path** key and select **Modify**.
	  b In the **Edit String** window, in the **Value** data: field, type the correct path from step 5.
	  c Click **OK**.
	 7 Close the **Registry Editor**.

Required for	Task Description
All	**3** Install this release in a test environment before installing on your production database.
	Create a test environment to review new features and for use after upgrading your production environment. If your installation includes customizations, you can use the test environment to compare your customizations with your upgraded production environment.
	Refer to the Sage SalesLogix Support Portal Web site for more information. Search for "creating a test environment" for information on using a copy of your production database in a test environment.
	Note Make sure you change your logging paths in the Administrator to access the test environment or errors will occur.
All	**4** Grant admin rights to the person installing the upgrade.
	Offline (disconnected) Web Client users can upgrade without admin rights on their machines. Other users require admin rights when installing Sage SalesLogix.
All	**5** Run the Sage SalesLogix Component Finder utility.
	The Sage SalesLogix Component Finder utility compiles a list of the Sage SalesLogix products installed on the machine.
	To run utility:
	1 In the **Upgrades** > **FindProducts** folder, double-click the **FindSalesLogixProducts.exe**.
	2 In the **Sage SalesLogix Component Finder** window, click the **Find SalesLogix Products** button.
Web	**6** Open Windows Firewall and add an exclusion for port 11211.
	The new Sage SalesLogix Cache Server requires access to this port. The Sage SalesLogix Cache Server is used to speed processing of the Web Client by providing storage for cached information.
	Do not install the Sage SalesLogix Cache Server on the same server as the SpeedSearch Server as both are resource intensive and will decrease performance. This is the default location and is recommended for optimum performance.
Protect Your Customizations	
Customized Network	**7** Record your Network customizations and use the Architect to create a bundle of all customized plugins in your database. See the "Working with Bundles" topic in the Architect Help for instructions.
	Note Sage SalesLogix does not overwrite your customized plugins. However, this bundle can be used as a backup of your customizations.
	Note Converting fields to Unicode is considered a customization for upgrade purposes because it is a change to the entity model; upgrades after you convert to Unicode may require that you bundle the Unicode model and merge it.
Customized Network	**8** Review the Sage SalesLogix plugin changes to determine if you have customized any plugins that are updated in this release.
	Review Chapter 3, "Plugin Changes in this Release" for all plugin changes. After reviewing plugin changes, determine if you are going to add your customizations to the plugins in this release (recommended) or add the Sage SalesLogix changes to your custom plugins.

Required for	Task Description
Customized Network and Customized Web	**9** Back up all custom assemblies to protect any changes. Custom assemblies may be overwritten or deleted during the installation.
This task is required if you are upgrading from Customized Web v7.5.0 **This task is not required if you are upgrading from** Customized Web v7.5.1, v7.5.2, v7.5.3, or 7.5.4	**10** Document the values for the Mashup Name and Result Name properties for all Timeline Visualization controls on each of your custom Web forms. The Timeline Visualization control has been upgraded to point to a Mashup Data Source control instead of directly to a mashup processor. Custom forms that contain Timeline Visualization controls must be updated manually. Before you upgrade, use the Application Architect to note the values for your Timeline Visualization controls. Then, task 57 describes how to apply the changes after the upgrade.
This task is required if you are upgrading from Customized Web v7.5.0 or v7.5.1 **This task is not required if you are upgrading from** Customized Web v7.5.2, v7.5.3, or 7.5.4	**11** Document or save any customizations made in the Hibernate.xml configuration files. The Hibernate.xml file has been deleted from all portals in the VFS Explorer and is now auto-generated from the template file (Default-hibernate-xml.configuration.codetemplate.xml). This template allows you to make customizations (generally for caching entities) in one location instead of using multiple Hibernate.xml files. Before you upgrade, use the Application Architect to note customizations in all Hibernate.xml files. Use the steps in task 59 to apply the changes after the upgrade. **Important** You must record your customizations before upgrading. When you open the Application Architect after upgrading, all Hibernate.xml files are deleted.
Customized Web	**12** Document or save any customizations made in the web.config files. Before you upgrade, use the Application Architect to note customizations in all web.config files. Use the steps in task 58 to apply the changes after the upgrade.
This task is required if you are upgrading Sage SalesLogix Mobile.	**13** Save any customization made to Sage SalesLogix Mobile. Before deleting the existing Sage SalesLogix Mobile portal, you must save any changes you have made directly in the portal SupportFiles to a bundle or onto your local file system. This includes changes to out-of-the-box views or classes, any views or JavaScript files you have added, or any changes to configuration or content. Failure to save these changes will cause them to be lost when you delete the portal.

Required for	Task Description
This task is required if you are upgrading from v7.5.0, v7.5.1, v7.5.2, or v7.5.3. **This task is not required if you are upgrading from** v7.5.4	**14** Delete the MergeConfiguration.xml and PotentialMatchConfiguration.xml files. **Note** If you customized these files, record your customizations before deleting them. New configurations have been added to the MergeConfiguration and the PotentialMatchConfiguration files. You must delete and recreate the file to see the updates. The files are rebuilt when you convert a lead in the Web Client. After you recreate these files, you can add your customizations. **To delete** 1 In the Application Architect, open the Virtual File System Explorer. 2 In the tree view expand **Configuration** and expand **Global**. 3 If necessary, save the **MergeConfiguration.xml** and **PotentialMatchConfiguration.xml** files to another location or record any customizations to the files. 4 Delete the files.

Customized Web with Hot Fixes and/or Accelerators	**15** Identify customized forms on your system and create a bundle of your Web customizations. Create a new project workspace and add the hot fixes and/or Accelerators (including the Mobile 2011R1 Accelerator) currently installed on your system.
	Note If you have not customized your Web implementation or have an accurate record of your customizations, disregard this task and proceed to task 21. If you customized your Web implementation, but do not have any hot fixes or Accelerators installed, proceed to task 16.
	
	To determine if a hot fix or Accelerator has been installed, view your system bundles:
	• For network bundles, open the Administrator, and then click Bundles.
	• For Web bundles, open the Application Architect, open Project Explorer, expand the VFS project workspace, and then expand Bundle Model.
	
	After creating a new project workspace, restore the project backup into the workspace, and apply your hot fixes and Accelerators.
	
	To create
	1 Create a new project workspace.
	  a On the Application Architect **View** menu, click **Project Workspace Manager**.
	  b Right-click in the **Project Workspaces** window, and then click **Add**.
	  c In the **Name** box, type a name for the project workspace.
	  d In the **Working Path** box, type or browse to the location where you want to save the new project workspace. The Source Path is automatically set to the working source location.
	  e In the **Description** box, type a description or additional information about the new project.
	  f Ensure the **Export Files Upon Creation** check box is not selected.
	  g Click **Create**.
	2 Restore the appropriate project backup to your project workspace.
	  a Locate the appropriate project backup file:
	    • **Sage SalesLogix v7.5 Project.Backup.zip** - when upgrading from v7.5.0.x.
	    • **Sage SalesLogix v7.5 SP1 Project.Backup.zip** - when upgrading from v7.5.1.x.

Required for	Task Description
Customized Web with Hot Fixes and/or Accelerators	**Task 15 - continued** • **Sage SalesLogix v7.5 SP2 Project.Backup.zip** - when upgrading from v7.5.2.x. • **Sage SalesLogix v7.5 SP3 (Cloud) Project.Backup.zip** - when upgrading from v7.5.3 Cloud. • **Sage SalesLogix v7.5 SP3 Project.Backup.zip** - when upgrading from v7.5.3 on-premise. • **Sage SalesLogix v7.5 SP4 Project.Backup.zip** - when upgrading from v7.5.4 on-premise. The project backup files are included in the service pack download files. b Copy the zip file to a local folder on the Application Architect computer. Ensure the folder path is as short as possible. c In the Application Architect Project Explorer, open the project workspace you created in step 1, right-click the project, and then click **Restore Project**. d In the **Select Project Backup File to Restore From** dialog box, browse to the project backup on your local machine (step b). 3 Apply all Web hot fixes and Accelerators currently installed in your production environment to the new project. 4 Proceed to task 17.
Customized Web Without Hot Fixes or Accelerators	**16** Prepare to create a bundle of your Web customizations. **Note** If you have not customized your Web implementation, disregard this task and proceed to task 21. If you customized your Web implementation and have hot fixes and/or Accelerators installed, proceed to task 17. Sage SalesLogix provides project backups that you can use to compare to your current database. A bundle is created from the differences. **To prepare** 1 Locate the appropriate project backup file: • **Sage SalesLogix v7.5 Project.Backup.zip** - when upgrading from v7.5.0.x. • **Sage SalesLogix v7.5 SP1 Project.Backup.zip** - when upgrading from v7.5.1.x. • **Sage SalesLogix v7.5 SP2 Project.Backup.zip** - when upgrading from v7.5.2.x. • **Sage SalesLogix v7.5 SP3 (Cloud) Project.Backup.zip** - when upgrading from v7.5.3 Cloud. • **Sage SalesLogix v7.5 SP3 Project.Backup.zip** - when upgrading from v7.5.3 on-premise. • **Sage SalesLogix v7.5 SP4 Project.Backup.zip** - when upgrading from v7.5.4 on-premise. The project backup files are included in the service pack download files. 2 Copy the zip file to a local folder on the Application Architect computer. Ensure the folder path is as short as possible. 3 In the Application Architect Project Explorer, open your current project. 4 If necessary, open the Output Window to display progress when creating your customization bundle. **Note** In the Project Explorer tree view do not expand nodes unnecessarily. Some nodes, such as the Portal Page Templates node, will autocreate items that may be included in your customization bundle. 5 Proceed to task 17.

Required for	Task Description

Customized Web	**17** Create a bundle of your Web customizations.
	Note If you have not customized your Web implementation, disregard this task and proceed to task 21.
	
	To create
	1 In Application Architect, open Project Explorer.
	2 Right-click **Bundle Model**, and then click **Create Manifest by Project Differences**.
	    The Select a Project to Compare Against dialog box opens.
	3 Choose one of the following:
	  • If your custom installation does not include hot fixes, select **Project Backup** and browse to the project backup you created in task 16.
	  • If your custom installation includes hot fixes and/or Accelerators, select **Existing Project**, and then select the project you created in task .
	4 Clear the **Include deleted items that were added to the current project** check box.
	    Clearing this check box ensures that items in the project's Recycle Bin are not included in the customization bundle as delete items.
	5 Click **OK**.
	    The new manifest opens in the Application Architect.
	6 On the New Manifest **Properties** tab, type a name in the **Name** box. For example, Customizations.
	7 Save the manifest.
	8 Expand **Bundle Model**, right-click the new manifest, and then click **Create Bundle**.
	9 In the **Save Bundle** dialog box, browse to the location where you want to save the bundle, type a name, and then click **Save**.
	    Note The bundle cannot be saved to a network drive. Save the bundle to a local drive or map the drive and save without using a UNC path.

This task is required if you are upgrading from Customized Web v7.5.0 or 7.5.1 **This task is not required if you are upgrading from** Customized Web v7.5.2, v7.5.3, or 7.5.4	**18** Save any customized Web Help files to a location outside the project. In v7.5.2, the Web Help was converted from .aspx files to .htm. When upgrading from v7.5.2 and later your customizations will be captured in the bundle process. However, when upgrading from v7.5.0 or v7.5.1 you must recreate your customizations in the new help topics. See the "Upgrading Customized Help" topic in the Developer Tips Help for details.

Required for	Task Description
Customized Web	**19** Run the BundleDiff utility using the "Common" parameter to preview how Sage changes impact your Web customizations. You can use Sage differences bundles with the BundleDiff utility to find differences between two bundle manifests. Use this information to determine conflicts between your customizations and Sage changes. Any files with conflicts must be merged into the upgraded environment to ensure it contains your customizations and Sage changes. The differences bundles are located in the Project Differences folder. The BundleDiff utility is installed in the \\Program Files\SalesLogix folder. The utility writes output to the console. You can redirect the output to a log file where you can open and save in text form. **BundleDiff Parameters** **Note** Utility parameters are case-sensitive. • **AllItems** - lists every item in both bundles. This is a flat list of bundle contents without differencing. You can use this to see a complete list of what items Sage changed in the release. • **Common** - lists the bundle items that exist in both bundles. These are the items that you will have to act on (ignore or merge) when you apply your customization bundle. • **Bundle1OnlyItems** - lists items that exist in bundle 1 but not in bundle 2. • **Bundle2OnlyItems** - lists items that exist in bundle 2 but not in bundle 1. • **CommonAndExclusive** - lists all items using the Common, Bundle1OnlyItems, and Bundle2OnlyItems parameters. **Example Parameter String** The following example compares two Sage SalesLogix upgrade bundles, lists the items that are common in the two bundles, and outputs the information to a text file on the local drive. "C:\Program Files\SalesLogix>bundlediff /B1:"c:\bundle754to80.zip" /B2:"c:\bundle80to80RC1.zip" /O:"common" > c:\commonout.txt".
This task is required if you are upgrading Sage SalesLogix Mobile.	**20** Remove previous versions of Sage SalesLogix Mobile. All prior versions of Sage SalesLogix Mobile should be removed before upgrading. Delete the Sage SalesLogix Mobile portal from the Sage SalesLogix Application Architect. **To delete** 1 In the Application Architect, open the **Project Explorer**. 2 Expand the **Portal Manager**. 3 Right-click the **Sage SalesLogix Mobile** portal and then click **Delete Portal**. 4 Click **Yes**. 5 Expand **Bundle Model**. 6 Right-click **Sage SalesLogix Mobile vX.X** and then click **Delete**. 7 Click **Yes**. 8 Click **Save All**. 9 Open Windows Explorer and browse to the deployment files. For example: …\inetpub\wwwroot. 10 Right-click **SlxMobile** and then click **Delete**.

Required for	Task Description
Prepare Your Sage SalesLogix Database	
Microsoft SQL Server	**21** If you are using Microsoft SQL Server and have published your database for replication, remove replication. Remember to reset replication after your upgrade is complete. **Note** Merge and Snapshot Replication are not compatible with Sage SalesLogix.
All	**22** Run the Integrity Checker on your database. Integrity Checker is run from the Tools menu in the Administrator. **Note** See the "Integrity Checker" topic in the Administrator Help for instructions. If you are running the Integrity Checker on a database that is not located on the same network as the SalesLogix Library and attachments, clear the Attachment and Library tests before you Repair the database.
All	**23** Make sure all users have logged off from Sage SalesLogix.
Remotes	**24** Instruct all Remote users and Offices to run a final synchronization cycle. **Note** Once the Main office is upgraded, remote users will still be able to synchronize changes to the host but will not be able to receive any changes.
Remotes	**25** Execute a final cycle of all sync servers. The final sync cycle must run successfully and send out all pending transactions.
Remotes	**26** Stop the Sync Service(s). If using a third-party scheduling package, deactivate the Synchronization Server tasks.
All	**27** Close all Sage SalesLogix applications and stop all applications and services that access the Sage SalesLogix database using the Sage SalesLogix OLE DB Provider.
All	**28** Back up your production database. Verify the backup is successful and complete. You may also want to create a backup at various points in the upgrade process. For example, after running the patch but before installing bundles. Then, if you need to revert your installation, you can choose a backup created during the upgrade without having to start from the beginning.
All	**29** Upgrade your main office computers using the Administrative Tools and Servers installation. This includes the Administrative Workstation, Synchronization Server, SpeedSearch Server, and Sage Job Server. **Note** The installation creates folders and applies permissions necessary for Sage SalesLogix to function. To avoid unexpected results, Sage recommends contacting Sage SalesLogix Professional Services Group or your Business Partner before changing default settings on these folders. The Administrative Tools and Servers installation allows you to select one or multiple applications to install on the same computer. The installations remove only the applications with a prior version and then install ONLY the applications for the installation that you selected. For example, if you have the Administrator, Architect, and Sage SalesLogix Client on your Administrative Workstation computer and you run the Administrative Tools and Servers install, only the Administrator and Architect will be removed and reinstalled. To upgrade the Sales Client, you must then run the Network Client install at which point only the SalesLogix Client will be removed and reinstalled. **Important** Sage Job Service is required for Web Client functionality.

Required for	Task Description
Upgrade your Main Office	
All	**Task 29 - continued**

To upgrade

1   Sage SalesLogix DVD > **Standard Server Installation** > **Required Administrative Tools and Servers**.

    The Express installation is for new implementations only and cannot be used to upgrade.

    Note If the installation does not detect the necessary prerequisites, you will be prompted to install them. Click Install to allow Sage to install the required components or Cancel to stop the installation.

2   In the **Sage SalesLogix Admin Tools and Servers** message box, click **Yes**.

3   On the **Welcome** and **License Agreement** screens, read the information and accept the agreement, and then click **Next**.

4   On the **Setup Type** screen, select an installation type, and then click **Next**. The following components are new in this release:

- **Sage SalesLogix SData Sync Service** - installs the service used for Accounting Integration functionality.

- **Sage SalesLogix Job Service** - installs the service used for certain Web Client functionality

5   Depending on your installation type, you may be prompted to enter some or all of the following information:

- **SQL Server sa password** - If you have a password set on the sa account on the SQL Server, type the sa password. The installation requires this password to install and attach the Sage SalesLogix databases.

- **SQL Server sysdba password** - Type your Microsoft SQL Server sysdba password. The installation must validate the password of the sysdba user to create a valid connection string for the Sage SalesLogix database.

- **Use Local System Account** - Select this option if the local user account has the correct security permissions to install Sage SalesLogix.
  If you clear this option, set the information for the Sage SalesLogix Service user.

- **Domain** - Type the network domain where you created the Sage SalesLogix Service user.

- **User Name** - Type the name of the Sage SalesLogix Service user (for example, SLXService).

- **Password** and **Confirm** - Type the Sage SalesLogix Service user's password.

- Port **Change** button - Changes the port number used for communication between the Clients and Sage SalesLogix Server. In most implementations, the default port number does not need to be changed. However, if you have another application or service using port 1706, you should change the port number to an unused port.

6   On the remaining screens, click **Install** and **Finish** to complete the installation.

    Note The installation process removes prior versions before installing the new applications. During this process, you may see the message "Assembly not Found." This is triggered between removing and replacing a service and does not indicate a problem with your installation. Click OK to continue the installation.

7   If prompted, restart your computer when the installation completes.

Required for	Task Description

Web	**30** Change Log On user for Sage Job Service.
	
	The Sage SalesLogix Job Service is installed with Admin Tools and Servers and requires the following:
	
	The Sage Job Service must be able to log on as a Sage SalesLogix user, and that user must have access to SLX Application Entities (such as Leads and Opportunities) and Activities. By default, the user assigned to the Sage Job Service is the Local System Account user. After installation, you must set the Job Service to log on as the WebDLL user after making sure the WebDLL user is mapped to the Sage SalesLogix Admin user. If your implementation includes Remote Office(s), ensure there is a trust relationship between the domain at the Main office and the domain at the Remote office.
	
	To change
	1 Open the Administrator.
	2 On the Navigation Bar, click **Users**.
	3 In the **Users** view, double-click the **Admin** user name, and then click the **General** tab.
	4 In the **Windows ID** box, click the Find button.
	   The Please select a Windows user to match the SalesLogix user dialog box appears.
	5 In the **Add Name** box, type your server name and user name, and then click **OK**.
	   For example, servername\WebDLL.
	6 In the **Import user data** box, click **No**, and then click **OK**.
	7 Set permissions to allow the WebDLL user to alter the DNS on the domain.
	8 Open Windows Services and right-click the **Sage Job Service**.
	9 Click **Properties**.
	10 In the **Sage Job Service properties** dialog box, click the **Log On** tab and select the **This account** option button.
	11 Click **Browse**.
	12 Specify the Location.
	   For example, the domain where you created the WebDLL user.
	13 Type the name of the domain user credentials specified in step 5.
	   In this example, the WebDLL user.
	14 Restart the Job Server and verify that it successfully started.

Required for	Task Description

All	**31** If the Sage SalesLogix Client is installed on your Administrative Workstation, run the Sage SalesLogix Network Client installation to install the Client application.
	
	To upgrade
	
	1 Sage SalesLogix DVD > **Install SalesLogix Client** > **Install Sage SalesLogix Network Client**.
	Note If the installation does not detect the necessary prerequisites, you will be prompted to install them. Click Install to allow Sage to install the required components or Cancel to stop the installation.
	2 In the **Sage SalesLogix Network Client** message box, click **Yes**.
	3 On the **Welcome** and **License Agreement** screens, read the information and accept the agreement, and then click **Next**.
	4 On the **Setup Type** screen, select an installation type, and then click **Next**.
	  • Click **Complete** to install the most common components.
	  • Click **Custom** to install only certain components or to change the installation location. Use the **Custom Setup** screen to enable or disable items for installation.
	5 On the remaining screens, click **Install** and **Finish** to complete the installation.
	If your company is not using Intellisync for SalesLogix to share information between Sage SalesLogix and Outlook, clear the Install Intellisync for Sage SalesLogix check box.
	Note The installation process removes prior versions before installing the new applications. During this process, you may see the message "Assembly not Found." This is triggered between removing and replacing a service and does not indicate a problem with your installation. Click OK to continue the installation.
	6 If prompted, restart your computer when the installation completes.

All	**32** Start the Administrator.
All	**33** Upgrade the saleslogix.sxc. The saleslogix.sxc is a list of all the checks the Integrity Checker runs. The file has been updated for Sage SalesLogix v8.0. The Integrity Checker, when opened, will create a new saleslogix.sxc if it cannot find one in the correct directory. **To update** 1 Browse to C:\ProgramData\SalesLogix\Integrity Checker for Windows 2008 or C:\Documents and Settings\All Users\Application Data\SalesLogix\Integrity Checker for Windows 2003. 2 Rename or delete the **saleslogix.sxc**. 3 In the **Administrator**, click on **Tools** and select **Integrity Checker**.

Required for	Task Description
Install the Service Pack Bundle	
All	**34** Use the Administrator to apply the bundle named SalesLogix v7.5x to v8.0 Upgrade.sxb. See the "Installing a Bundle" topic in the Administrator Help for instructions.

Note If you are running Sage SalesLogix v7.5.1, 7.5.2, 7.5.3, or 7.5.4 on Oracle, clear the Create Field [VirtualFileSystem:IsCompressed VarChar(1) NULL] statement in the v7.5.1 section of the bundle during installation. This statement causes an Oracle database to stop responding. If you are upgrading from v7.5.0, do not clear this statement.

The bundle is located in the Upgrades folder. The bundle contents are outlined in Chapter 3, "Plugin Changes in this Release". This includes functionality formerly released in hot fixes.

Understanding the Conversion Utility
Note You must log on to the conversion utility as admin.

The conversion utility runs from the upgrade bundle and converts information for sales orders, Send SLX attachments, SpeedSearch, and User Feature Security. The conversion utility can be run at a later time using the SLX75SP2Conversion.exe in the \\Program Files\SalesLogix folder.
Note If you ran the conversion utility in an earlier upgrade and converted all items, you do not need to run the conversion a second time. You can clear any items you have already converted or cancel the conversion if all items have been converted.

Fix Security Profiles Conversion

The conversion reads the total number of offsets (used by Field Level Security) from the SecTableDefs table and updates the string length in the SecProfiles table ProfileData field with the correct string length.

Sales Orders Conversion

The conversion performs the following to the SalesOrder table:
- Creates a sales order address for each sales order (all sales orders must contain a Billing and Shipping address). Previously the Network Client did not create the SalesOrderAddresses until the sales order was closed which could cause compatibility issues with the Web Client. To resolve any compatibility issues, the conversion validates that every sales order contains both a Billing and Shipping address. If it does not and the sales order is not Closed, then the BillingAddressID and ShippingAddressID will join to the Address table to get the appropriate record and write it to the SalesOrderAddress table.
- Updates the OrderTotal field by calculating each of the sales order items using Price * quantity - discount.
- Updates the GrandTotal field. This field allows sales order groups to display the sales order grand total without performing a function to retrieve the data. GrandTotal is calculated using OrderTotal - discount + freight + tax.
- Updates the AccountManager field based on the sales order associated with the opportunity's account manager. If the sales order is not associated with an opportunity, the sales order account association is used.

User Feature Security Conversion

The conversion changes the data in the UserFeatureSecurity table FamilyPluginName field. This field currently stores the PluginID but after the conversion will store the plugin Family and Name.
Note If you have custom code that references this field, you may need to update it after this change.

Required for	Task Description
All	**Task 34 - continued**

Update Attachment Counts Conversion

The conversion gets and sets the AttachmentCount field in the Activity and History tables based on the number of attachments in the each table.

Default SpeedSearch Indexes Conversion

The conversion updates the Ticket, Defect, Ticket Internal, and Defect Internal out-of-the-box indexes (that have not been edited) to include the Alternate Key (Displayed ID). After the conversion, SpeedSearch returns the Alternate Key (Ticket Number or Defect Number) instead of the TicketID or DefectID when a customer does not have access to the record.

Send SLX Attachments Conversion
Note Microsoft Outlook or MAPI must be installed on the computer where you run the Send SLX Attachments conversion.

The conversion enables you to choose how Send SLX handles attachments. This setting is also available in the Administrator > Tools > Options > Outlook tab.

The Send SLX button in Outlook saves e-mail attachments using one of the following methods:

Option 1: Bundle attachments with e-mail message (.msg file)
- Functionality in v7.5 and later
- A single e-mail attachment contains the e-mail and all attachments
- If the user clicks "Yes" when prompted to save attachments to the account or contact that is associated to the e-mail, the e-mail message and all attachments are saved as a single attachment.

Option 2: (Recommended) Save attachments as individual files
- Functionality in versions prior to 7.5.0
- E-mail attachments are saved as individual files
- If the user clicks "Yes" when prompted to save attachments to the account or contact that is associated to the e-mail, the individual files are saved as attachments.
  - **Convert MSG files associated to e-mail history records** - select this option to convert existing .msg files.
    - .msg files associated to an e-mail history record will be converted to individual attachments. The conversion removes the attachments from the .msg file but leaves the .msg file containing only the e-mail message with the history record.
    - .msg files that were attached manually (not using Send SLX) are not converted.
    - .msg files attached to Open activities are not converted.
  - **Convert files on history records created after** - select this option to convert files associated to history records created after the specified date.

Exclude graphics files - select this option and specify any files you want to exclude based on type and/or size. For example, you may want to use this option if your company uses graphics files in e-mail signatures and you do not want to add them as attachments.

Required for	Task Description
Oracle	**35** Instruct your Oracle DBA to run the CreateOracleViews.sql script.
	Running this script allows you to create necessary Sage SalesLogix views without extra permissions for the sysdba user.
	Before running the script, open the script in Notepad and document any changes impact your customizations. Ensure you reapply your customizations after running the script. **Note** The script must be run as the sys user.
	To run
	1 Ensure you have applied the SalesLogix v7.5x to v8.0 Upgrade.sxb bundle (task 34). The CreateOracleViews script requires tables added in the service pack bundle.
	2 Open Oracle SQL *Plus Worksheet or SQL Developer.
	3 On the **File** menu, click **Open**.
	4 Browse to the **CreateOracleViews.sql** script.
	The script is located in the Oracle folder.
	5 To load the script into the Query section, click **Open**.
	6 To run the script, click **Execute**.
Oracle	**36** Instruct your Oracle DBA to run the OracleFixes.sql script.
	Running this script alters the TargetResponse table. **Note** The script can be run as the sys or sysdba user.
	To run
	1 Open Oracle SQL *Plus Worksheet or SQL Developer.
	2 On the **File** menu, click **Open**.
	3 Browse to the **OracleFixes.sql** script.
	The script is located in the Oracle folder.
	4 To load the script into the Query section, click **Open**.
	5 To run the script, click **Execute**.
Reapply your Network Customizations	
Customized Network	**37** Apply the changes listed in Chapter 3, "Plugin Changes in this Release". You can use one of the following strategies: • Add your customizations to the Sage SalesLogix plugins (recommended). • Add the Sage SalesLogix changes to your custom plugins.
	After applying plugin changes, release the appropriate plugins using Architect. **Note** See the "Release a Plugin for Use" topic in the Architect Help for information on releasing plugins.
Upgrade Synchronization	
Remotes	**38** Cycle your Synchronization Server(s) so that changes are sent to remotes. Use Custom Sync since it is not necessary to run subscription. **Note** See the "Creating a Custom Sync Cycle" topic in the Administrator Help for more information.
All	**39** Restart the Sync Service(s).

Required for	Task Description
Upgrade Network and Remote Users/Offices	
All	**40** Run the Sage SalesLogix Network Client installation on each of your Network user's computers. You can install the Sage SalesLogix Client using one of the following methods. • **Build an automated installation**. Automated installations create an executable file that allows users to install with no user intervention. Refer to the *Sage SalesLogix Implementation Guide* for details on building automated installations. • **Run the Sage SalesLogix Client installation**. The Sage SalesLogix Client can be installed on each user's computer using the Client installation. If you plan to run a custom installation, you must install the Client using the standard installation. See task 31 for detailed steps.
Remote Office only	**41** Remove the Intellisync Portal from IIS and the folders from the file system. Intellisync must be removed from the remote office before upgrading. **To delete** 1 Open IIS. 2 Expand the **SalesLogix** web site. 3 Right-click the **SlxIntellisync** portal and select **Delete**. 4 Close IIS. 5 Browse to C:\inetpub\wwwroot. 6 Right-click the **Intellisync** folder and select **Delete**. 7 Reset IIS.
Remotes	**42** Distribute Client installations to your Remote users and Offices. You can distribute a copy of the Sage SalesLogix DVD, or the files and folders associated with an automated installation. If you created an automated installation of the Sage SalesLogix Client and the user is disconnected from the network, you must distribute the entire folder structure created during the automated installation to the remote computer. If the user has network access, copy the installation to a shared network drive.

Required for	Task Description

Remotes	**43** Direct each Remote Office to upgrade using the Remote Office installation.
	
	To upgrade
	
	1 Sage Saleslogix DVD > **Standard Server Installation > Remote Office Server**.
	Note If the installation does not detect the necessary prerequisites, you will be prompted to install them. Click Install to allow Sage to install the required components or Cancel to stop the installation.
	2 In the **Sage SalesLogix Admin Tools and Servers** message box, click **Yes**.
	3 On the **Welcome** and **License Agreement** screens, read the information and accept the agreement, and then click **Next**.
	4 On the **Setup Type** screen, select an installation type, and then click **Next**.
	5 Depending on your installation type, you will be prompted to enter some or all of the following information on the installation screens:
	  • **Use Local System Account** - Select this option if the local user account has the correct security permissions to install Sage SalesLogix. If you do not select this option, set the information for the Sage SalesLogix Service user.
	  • **Domain** - Type the network domain where you created the Sage SalesLogix Service user.
	  • **User Name** - Type the name of the Sage SalesLogix Service user (for example, SLXService).
	  • **Password** and **Confirm** - Type the Sage SalesLogix Service user's password.
	  • Port **Change** button - Changes the port number used for communication between the Clients and Sage SalesLogix Server. In most implementations, the default port number does not need to be changed. However, if you have another application or service using port 1706, you should change the port number to an unused port.
	  • **Configure Web Server** - This option is available for Web Remote Office installations. Clear this option for non-Web Remote Offices.
	  • **Create Web Site** – Select this check box if you want the installation to automatically create and deploy an Intellisync for SalesLogix Web Site.
	    • **Enter a web site name** – Type the name of the Web site. The default name is *SalesLogix*.
	    • **Port** – Enter the port number that the *SalesLogix* Web site will use.
	    • **Domain** – Enter the network domain where you created the WebDLL user.
	    • **User Account** – Enter the name of the WebDLL user you created. The default value is *WebDLL*. If you used a different name, type it exactly as you did when you created it.
	    • **Password** and **Confirm Password** – Type the WebDLL user's password.
	6 On the remaining screens, click **Install** and **Finish** to complete the installation.
	Note The installation process removes prior versions before installing the new applications. During this process, you may see the message "Assembly not Found." This is triggered between removing and replacing a service and does not indicate a problem with your installation. Click OK to continue the installation.
	7 If prompted, restart your computer when the installation completes.

Required for	Task Description

Remotes	**44** Direct Remote users to upgrade the Sage SalesLogix Client.
	
	Remotes should upgrade using the Sage SalesLogix Remote Client installation on the Sage SalesLogix DVD.
	
	To upgrade
	
	1 Sage SalesLogix DVD > **Install SalesLogix Client** > **Install Sage SalesLogix Remote Client**.
	
	Note If the installation does not detect the necessary prerequisites, you will be prompted to install them. Click Install to allow Sage to install the required components or Cancel to stop the installation.
	
	2 In the **Sage SalesLogix Admin Tools and Servers** message box, click **Yes**.
	
	3 On the **Welcome** and **License Agreement** screens, read the information and accept the agreement, and then click **Next**.
	
	4 On the **Setup Type** screen, select an installation type, and then click **Next**.
	  • Select **Complete** to install all components on this computer.
	  • Select **Custom** to install only certain components or to change the installation location. Use the **Custom Setup** screen to enable or disable items for installation.
	
	   If your company is not using Intellisync for SalesLogix to share information between Sage SalesLogix and Outlook, clear the Install Intellisync Web Portal check box.
	
	5 On the remaining screens, click **Install** and **Finish** to complete the installation.
	
	Note The installation process removes prior versions before installing the new applications. During this process, you may see the message "Assembly not Found." This is triggered between removing and replacing a service and does not indicate a problem with your installation. Click OK to continue the installation.
	
	6 If prompted, restart your computer when the installation completes.

Web Remotes	**45** Instruct Offline Web Client users to upgrade. Users do not need administrator rights to install or run the Offline Web Client. However, the Offline Web Client installation installs third-party prerequisites if they are not detected on the user's computer. Administrator rights are required to install the following: • Windows Installer 4.5 • Windows Powershell 1 • Microsoft SQL Server 2008 R2 • .Net Framework 4.0 or later • Microsoft SQL 2005 Backwards Compatibility • .Net AJAX Extensions • Internet access to Microsoft.com You can grant the user installing the Offline Web Client administrator rights or install all the required prerequisites before running the Sage SalesLogix installation. **Installing Without Administrator Rights** To install the Offline Web Client without administrator rights, ensure you have installed all the prerequisites listed in the previous section. If the Offline Web Client installation does not detect the necessary prerequisites and the user running the installation does not have administrator rights, the Offline Web Client installation will not complete.

Required for	Task Description
Web Remotes	**Task 45 - continued** **Installing With Administrator Rights** To install the Offline Web Client and the required prerequisites, you must grant the user administrator rights for the computer on which you are installing Sage SalesLogix. See your Microsoft documentation for instructions on granting administrator rights using one of the following options: • Add each user to the local machine's Admin group for the duration of the implementation. • Use a domain user account that is a member of the domain admin group for the duration of the implementation. **To install** 1 Sage SalesLogix DVD > **Install SalesLogix Client** > **Install Sage SalesLogix Offline Web Client**. 2 On the **Welcome** and **License Agreement** screens, read the information and accept the agreement, and then click **Next**. 3 On the remaining screens, click **Install** and **Finish** to complete the installation. The following option may appear on the last screen: • **Install Intellisync for Sage SalesLogix** - If your company is not using Intellisync for SalesLogix to share information between Sage SalesLogix and Outlook, clear the Install Intellisync for Sage SalesLogix check box.
Remotes	**46** Instruct Remote users and Offices to log on to the Synchronization Client and sync immediately after upgrading. **Note** Ensure Remote users and Offices enable "Apply Changes" during the synchronization cycle.

All Web	**47** Upgrade the Web Host.
	
	To upgrade
	
	1 You must log on as the local administrator to install the Web components on each machine.
	2 On the **Sage SalesLogix Installation** screen, click **Standard Server Installation**.
	3 On the **Standard Server Installation** screen, click **Web Host on IIS**.
	  If the installation does not detect the necessary prerequisites, you will be prompted to install them. Click Install to allow Sage to install the required components or Cancel to stop the installation.
	4 In the **Sage SalesLogix Web Host** message box, click **Yes**.
	5 On the **Welcome** and **License Agreement** screens, read the information, accept the agreement, and then click **Next**.
	6 On the **Setup Type** screen, select an installation type, and then click **Next**.
	  • Select **Complete** to install the Web Host and SalesLogix Cache Server.
	  • Select **Custom** to install only certain components or to change the installation location. Use the **Custom Setup** screen to enable or disable items for installation.
	    If you install the SalesLogix Cache Server on a computer other than the Web Host, you must manually configure the web.config file to access the SalesLogix Cache Server computer. See the "Configuring the SalesLogix Cache Server" topic in the Application Architect help for details.

Required for	Task Description
Upgrade Your Web Workspace	
All Web	**Task 47 - continued**

Task 47 - continued

7 Depending on your installation type, you will be prompted to enter some or all of the following information on the installation screens:

- **Use an existing web site** - If you want the installation to use an existing Web site, click this button and select the Web site name. If you created a Web site for Intellisync for SalesLogix in the Administrative Tools and Servers installation, you can select the existing SalesLogix Web site for the Web components.
- **Create Web Site** – Select this check box if you want the installation to automatically create a Web Site. Selecting this check box creates a Web site you can use to deploy the Web portals.
- **Enter a Web site name** – Type the name of the Web site that will appear in IIS. The default name is *SalesLogix*.
- **Port** – Enter the port number that the *SalesLogix* Web site will use. Note the port number for future reference. You must use port number 1025 or greater.
- **Domain** – Enter the network domain where you created the WebDLL user.
- **User Account** – Enter the name of the WebDLL user you created. The default value is *WebDLL*. If you used a different name, type it exactly as you did when you created it.
- **Password** and **Confirm Password** – Type the WebDLL user's password.
- **Configure SalesLogix Database** - Clear this check box if you do not want to connect to the Sage SalesLogix database, but still want to create a Web site. When you clear this check box, the other options on this screen are disabled.
  - **SalesLogix Server** – Select or type the name of the computer on which you installed the Sage SalesLogix Server.
  - **SalesLogix Database Alias** – Select or enter the name of the connection to the Sage SalesLogix database.
  - **Search for servers on port** - Enter a port. By default, this is set to port number *1706*.
  - **User Name** – Enter a Sage SalesLogix Server user name (such as Admin).
  - **Password** – Enter the Sage SalesLogix Server password for the SalesLogix user.

8 On the remaining screens, click **Install** and **Finish** to complete the installation.

Note The installation process removes prior versions before installing the new applications. During this process, you may see the message "Assembly not Found." This is triggered between removing and replacing a service and does not indicate a problem with your installation. Click OK to continue the installation.

9 If prompted, restart your computer when the installation completes.

Required for	Task Description
All Web	**48** Open the Application Architect and update existing projects. All existing projects that you open in the Application Architect must be updated to the new v8.0 structure. This only changes the format of the project.xml file and some of the XML components. You must still restore the Sage SalesLogix v8.0.0 Project.Backup.zip in task 49. When opening a project created prior to version 8.0, you may be prompted to update the project format. **To update** 1 Open the Application Architect. 2 The Application Architect automatically opens the last project you were working with and the **Confirm Upgrade** dialog box opens. 3 Click **Yes**. If you do not want to update the format of this project, and you want to choose a different project, click No. Then, open the project you want to update. 4 After updating the project, information about the update can be found in the Output Window.

All Web	**49** Restore the Sage SalesLogix v8.0.0 project backup to the VFS workspace.
	
	Sage SalesLogix provides a project backup file that contains all Web changes for this release. You must add these changes to upgrade your Web installation. When upgrading with project backups, you do not need to apply Web upgrade bundles.
	
	To guarantee that a client will always have access to the project model, deploy the client portal from a virtual file system-based project. It will always be accessible to the client because the VFS resides in the SalesLogix database. Remotes MUST be deployed from a VFS-based project, and it is recommended that production systems be deployed from a VFS-based project as well.
	
	To restore
	1 Locate the **Sage SalesLogix v8.0.0 Project.Backup.zip** file.
	    Project backups are located in the Upgrades\Project Backups folder.
	    Note Additional project backups are available for use when creating a customization bundle (task 16). Use the Sage SalesLogix v8.0.0 Project Backup file to receive all changes in this release. Disregard all other project backups when completing this task.
	2 Copy the zip file to a local folder on the Application Architect computer. Ensure the folder path is as short as possible.
	3 Open the Application Architect.
	4 In the Project Workspaces window, select the VFS project workspace.
	5 Right-click the project, and then click **Restore Project**.
	6 In the **Select Project Backup File to Restore From** dialog box, browse to **Sage SalesLogix v8.0.0 Project.Backup.zip** on your local machine (step 2).
	7 Click **OK**.

Required for	Task Description

SLX Job Server	**50** Add SageJobServer target and SlxJobService portal.
	
	The Sage Job Server enables you to schedule single and recurring tasks for immediate or delayed execution, either manually or dynamically at run-time. It executes tasks out of process, releasing worker threads and memory for client sessions.
	
	Sage SalesLogix Web Client functionality that requires the Sage Job Server includes:
	• Roll over incomplete activities
	• Update remaining days for active contracts
	• Update opportunities
	• Update leads
	• Delete leads
	
	To add
	1 In the **Deployment Explorer**, right-click **Core Portals** and select **Edit Deployment**.
	2 Right-click **Deployment Targets** and select **Add Target(s)**.
	3 In the **Select Target(s)** window:
	    a In the **Select Deployment Target(s)** area, select the **File System** check box.
	    b In the **Select Portal(s)** area, select the **SlxJobService** check box.
	    c Click **OK.**
	4 In the **File System Target Settings** area, browse to the location for the SlxJobService portal to be deployed to or enter the default path **%ALLUSERSPROFILE%\Sage\Scheduling\Tenants**. For example, C:\ProgramData\Sage\Scheduling\Tenants.
	Note You can change the location path from the default setting, however you must also edit the Sage.Scheduling.Server.exe.configedit to reflect the change. See An Introduction to Job Server in Sage SalesLogix.pdf for information on how to complete this task.

All web	**51** Build and deploy the portals. **To build and deploy** 1 On the **Build** menu, click **Build Web Platform**. 2 Click the **Deploy All** button when the build has completed.
Intellisync on Remote Offices	**52** Intellisync must be deployed from the Application Architect on the Sage SalesLogix host server. Once the portal has been deployed, run the Synchronization Server and the Sync Client. The registration service on the Remote Office will automatically install and configure the Intellisync portal for the Remote Office. **To deploy** 1 Expand **Deployments** and select the **Remote Sales Client** deployment. 2 In the **Deployment Targets** area, under **Remote Office(s)**, select the **SlxIntellisync** portal. 3 Click the **Deploy** button. 4 Run the Synchronization Server. 5 Instruct the remote offices to run the Sync Client.

Required for	Task Description
All Web on Windows 2003	**53** Update your portal configuration. **Note** For IIS7 and above, the Application Architect automatically configures the necessary portal settings. Sage SalesLogix now uses the .NET 4.0 framework so the ASP.NET version must be updated to reflect this change. **To update** 1 In IIS 6, expand your Web server machine, then expand **Web Sites**. 2 Expand the **SalesLogix** Web site (or the Web site where you deployed the SData portal), then right-click the **SData** portal and select **Properties**. 3 On the **ASP.NET** tab, select **4.0.30319** in the **ASP.NET version** drop-down box. 4 Click **Apply**. 5 Click **OK**. 6 Repeat the steps for all portals except Web Reporting. 7 Restart IIS.
All Web	**54** Close the Application Architect, stop and restart all Sage SalesLogix services, and reset IIS. **Note** If you have not customized your Web implementation, proceed to task 72.
	55 Use the Application Architect to apply your custom bundle. **Note** If you have not customized your Web implementation, disregard this task. Install the bundle you created of your customizations (task 17) to the new v8.0 project workspace. During the bundle installation, you can merge your customizations with the Sage changes. You can also validate the changes if there are duplicate items between the current project and the bundle you are installing. **Note** The default behavior is to overwrite the item in the target project (the 8.0 Sage SalesLogix version of the item) with your customization (from the customization bundle). After a merge, the modified file is the one that will be installed. It is not recommended to modify both files during a merge. However, if both files are modified, the file on the left is kept (unless your differencing tool specifies otherwise). **Upgrading Customizations Technical Preview** The QuickFormDiff utility is available as a technical preview in this release. (You can download this utility from the SLXToolsContrib repository at http://www.github.com/SageSalesLogix.) This utility provides a GUI interface that simplifies the process of merging your customizations with Sage changes. If you follow the process documented in the *QuickFormDiff Utility Readme.doc*, you do not need to merge changes when applying your customization bundle as documented in this task. If you are using the QuickFormDiff Utility Readme with this Upgrade Workplan, install your customization bundle and overwrite all Sage items. **Note** When you perform a multiple version upgrade (for example, from 7.5 to 8.0, you only need to merge customizations one time. This only applies to customizations, not other tasks.

Required for	Task Description
Reapply Your Web Customizations	

Customized Web	**Task 55 - continued**
	
	To merge changes during bundle installation
	
	1 In the Project Explorer, right-click your project folder, and then click **Install Bundle**.
	    The Select Bundle to Install dialog box opens.
	
	2 Browse to and select the bundle you created in task 17 containing your customizations, and then click **Open**.
	    The Install Bundle dialog box appears.
	
	3 Verify the **Bundle Path** information is correct, and then click **Next**.
	    The Optional Merged Content Path from a Previous Install option allows you to continue resolving conflicts from a prior bundle installation. If you have extensive customizations, you may have a large number of conflicts that must be resolved when applying your custom bundle.
	    If you use Preserve Merged Content (step 6), you can resolve conflicts over an extended period of time. For example, you may be installing a customization bundle with 50 conflicts and you have resolved only 25 of the conflicts. If you click the Preserve Merged Content button, cancel the installation, then apply the same bundle at a later time, use the Optional Merged Content Path to browse to the folder with merged content. Then, you can continue to resolve the remaining 25 conflicts from your customization bundle.
	
	4 On the **Select Items** screen, ensure the **Entity Model** and **Portals** check boxes are selected.
	    This screen indicates the number of items that conflict with existing items in the current project.
	
	5 Review the Support Files Bin folder for each portal application in your bundle and clear any DLLs that begin with Sage. You must select the item, and then clear the check box.
	    Assembly support files that are developed by Sage may be included in your customization bundle. These files are not necessary for the merge process.
	
	6 If you want to save changes made during the bundle installation, click the **Preserve Merged Content** button.
	    By default, all changes are saved to a temporary folder, and then deleted after the bundle installation is complete. When you preserve merged content, the folder containing merged customizations is not deleted. If you cannot resolve all conflicts at one time, click this button and use the Optional Merged Content Path from a Previous Install option when reapplying the bundle (step 3).
	
	7 Clear the **Only stop at items that require user intervention** check box if you want to view only duplicate items in the bundle.
	    By default, when this check box is selected and you move through the items in the tree view, the cursor stops only on items that require user intervention (such as determining which item to keep or how to merge duplicate items). The behavior at install for the duplicate items is controlled by the selected install action.

Required for	Task Description
Customized Web	**Task 55 - continued** 8 Use the **Next Dup** and **PrevDup** buttons to select and resolve any duplicate items. Items in red in the tree view indicate duplicates. These duplicates represent customized items in your bundle that are also in the project to which you are installing. A duplicate item with an install action set to DiffMerge requires the item to be merged before continuing the installation. 9 In the **Install Action** drop-down list, select **DiffMerge**. DiffMerge allows you to use a third-party differencing tool to compare a customization in your bundle with the duplicate customization in your project, and to merge the differences between the two. If you do not have a third-party differencing tool configured, you are prompted to configure one. 10 After making your changes, click **Merge Item**. The Merge Item button is enabled only when the DiffMerge install action is selected. If you clear a check box, the changes to that item will not be installed. **Note** Some bundle items have multiple files (resource files). Your differencing tool will display them twice; once for the main file, and once for the resource (.resx) file. If there are no changes in the linked .resx file, it does not display. 11 (Optional) To modify an item in the Bundle Model tree view without having to use a differencing tool again, click an item in the tree view, and then click **View Item**. The View Bundle Item text editor opens. **Note** If you need to remove all changes you have made to an item, click Undo Changes. This removes all changes and reverts to the original state of the item in the bundle 12 Click **Next**. 13 Click **Finish**.
Customized Web	**56** If necessary, apply your custom schema to the database. If the database you are restoring the project to is the database where your customizations were created, custom schema is already included. If you are applying a customization bundle that relies on custom schema to a new or different database than the database where the schema was developed, then you must also apply the bundle that contains your custom schema to the new database. You need to apply your custom schema if you created your customization bundle on a different database than the one on which you are applying the customization bundle.
This task is required if you are upgrading from Customized Web v7.5.0 **This task is not required if you are upgrading from** Customized Web v7.5.1, v7.5.2, v7.5.3 or v7.5.4	**57** Update Timeline Visualization controls on your custom forms. **To update** 1 In the Application Architect, open a custom form that contains a Timeline Visualization control. 2 Use each Mashup Name/Result pair you recorded in task 10 to create a Mashup Data Source control on the form. 3 For the Timeline Visualization **Data Source** property, set a value by selecting one of the Mashup Data Source control names from the drop-down list. 4 Save the form. 5 Repeat these steps for each custom form that contains a Timeline Visualization control.

Required for	Task Description
Customized Web	**58** Reapply customizations from the web.config file(s) to the web.config and appSettings.config file(s). Application settings have been moved from the web.config file to the appSettings.config file. Customizations to the appSettings section must be merged into the appSettings.config file. For more information see the "Editing Configuration Files" topic in the Application Architect help. **To update** 1 In the Application Architect, open the web.config file. 2 Update the new web.config files with the customizations you recorded in task 12. 3 In the Application Architect, open the appSettings.config file. 4 Merge any customization made to the appSettings section of the web.config file to the appSettings.config file.
This task is required if you are upgrading from Customized Web v7.5.0 and v7.5.1 **This task is not required if you are upgrading from** Customized Web v7.5.2, v7.5.3 or v7.5.4	**59** Reapply customizations from the Hibernate.xml configuration file(s) to the Default-hibernate-xml.configuration.codetemplate.xml file. **To update** 1 In the Application Architect, open the Default-hibernate-xml.configuration.codetemplate.xml file. The file is stored in Model > Entity Model > Code Templates > Entity 2 Use the customizations you recorded in task 11 to update the template.
This task is required if you are upgrading from Customized Web v7.5.0 and v7.5.1 **This task is not required if you are upgrading from** Customized Web v7.5.2, v7.5.3 or v7.5.4	**60** Update all custom pages that were generated as straight application pages (except entity pages and main views) to inherit from Sage.Platform.WebPortal.WebPortalPage. **To update** 1 In the Application Architect, open a custom application page. 2 In the Properties pane, expand **Misc**, and add the following class to the **Inherits From** property: Sage.Platform.WebPortal.WebPortalPage 3 Save the form. 4 Repeat these steps for all custom application pages.

Required for	Task Description

Customized Web	**61** Update any custom quick forms where the Currency Control ExchangeRateType value is set to OpportunityRate or SalesOrderRate.
	
	In earlier releases, the Currency Control ExchangeRateType contained four enum values (Opportunity Rate, Sales Order Rate, Base Rate, My Rate). In this release, ExchangeRateType contains three values (Base Rate, My Rate, Entity Rate) and two deprecated values (Sales Order Rate, Opportunity Rate). For custom forms where the ExchangeRateType value is set to either SalesOrderRate or OpportunityRate it is advisable to change to EntityRate.
	
	To update
	
	1 Open the xml file that contains the quick form configuration metadata.
	
	The xml files will be listed in the build exception. For example, ...\Model\Entity Model\SalesLogix Application Entities\SalesOrderItem\QuickForms\ EditSalesOrderItem.main.quickform.xml
	
	2 Find the ExchangeRateType node inside the ControlDefinition node for the currency control and change the value of the ExchangeRateType property from SalesOrderRate OR OpportunityRate to EntityRate.
	
	The following is an example of an updated ControlDefinition node:
	
	• Before:
	
	`<SalesLogix:Currency runat="server" ID="curSubTotal" DisplayMode="AsText" ExchangeRateType="SalesOrderRate" DisplayCurrencyCode="true">`
	`</SalesLogix:Currency>`
	
	• After:
	
	`<SalesLogix:Currency runat="server" ID="curSubTotal" DisplayMode="AsText" ExchangeRateType="EntityRate" DisplayCurrencyCode="true">`
	`</SalesLogix:Currency>`

Customized Web	**62** If you have custom code, review the list of breaking changes to the Sage SalesLogix API and change references to classes and members in your custom code as appropriate. See the SDK Documentation area on SupportOnline for the API Change List for this and previous releases.

Required for	Task Description

This task is required if you are upgrading from Customized Web v7.5.0 and v7.5.1
 	
This task is not required if you are upgrading from Customized Web v7.5.2, v7.5.3 or v7.5.4
	63 Update all custom smart parts that display in a dialog workspace.
	Note Custom quick forms are automatically updated and do not require manual changes.
	
	If you have a custom smart part that displays in a dialog box, you must update the close event to call the IPanelRefreshService.RefreshAll() method. If a smart part is not updated, any changes made within the dialog box will not show in the form behind the dialog box until the user manually refreshes.
	
	To update
	1 Determine which custom smart parts you need to update.
	
	You can run a global search on the CloseEventHappened() method to return a list of custom smart parts to review and possibly change. This search may not return a complete list of custom smart parts, but can be used as a starting point.
	
	2 Use one of the following methods to update the smart part to call the Refresh method.
	Method 1: Call Refresh within save/update logic
	<pre>protected void cmdOK_Click(object sender, EventArgs e)
	{
	// ... save/update logic ...
	// Close dialog
	DialogService.CloseEventHappened(sender, e);
	// Refresh the page
	Refresh();
	}</pre>
	Method 2: Wire up a Refresh event handler (also newly added to SmartPart class)
	<pre>protected override void OnWireEventHandlers()
	{
	base.OnWireEventHandlers();
	cmdOK.Click += cmdOK_ClickAction;
	cmdOK.Click += DialogService.CloseEventHappened;
	cmdOK.Click += Refresh;
	cmdCancel.Click += DialogService.CloseEventHappened;
	}</pre>

Required for	Task Description

This task is required if you are upgrading from Customized Web v7.5.0 and v7.5.1
 	
This task is not required if you are upgrading from Customized Web v7.5.2, v7.5.3 or v7.5.4
	64 Reapply customizations to the Activity Notes and LongNotes fields.
	
	In this release, both Notes and LongNotes are exposed as properties. The LongNotes property ensures that the Notes property always contains the first 255 characters of what is set in the LongNotes property.
	
	If you have customizations with text boxes bound to the Notes property, the LongNotes field will not be updated when values in these text boxes are changed. To keep data synchronized in the Notes and LongNotes fields, update your customizations to bind controls to the LongNotes property.
	Note The Notes field should be used in data grids. This makes sorting possible in that column. Use LongNotes for two-way binding (for example, to a text box) and use Notes for one-way/ReadOnly binding (for example, to a data grid column).
	• If you have customized any forms or quick forms that are bound to Activity.Notes, update them to bind to Activity.LongNotes.
	Change the following text:
	`bs.Bindings.Add(new WebEntityBinding("Notes", Notes, "Text"));`
	to this:
	`bs.Bindings.Add(new WebEntityBinding("LongNotes", Notes, "Text"));`
	• If you have customized any business rules or other code that assigns values to Notes, update them to assign values to the LongNotes property.
	Change the following text:
	`newHistory.Notes = lead.Notes;`
	to this:
	`newHistory.LongNotes = lead.Notes;`

This task is required if you are upgrading from Customized Web v7.5.0 and v7.5.1 **This task is not required if you are upgrading from** Customized Web v7.5.2, v7.5.3 or v7.5.4	**65** Update customizations that reference the Add Edit Address view for sales orders. The new Add Edit Sales Order Address view is called when editing a sales order address. This view binds to a sales order address where it previously was bound to a standard address. If you have customized code that references the standard address from a sales order, you may want to update that code to reference the sales order address.

Required for	Task Description
This task is required if you are upgrading from Customized Web v7.5.0 and v7.5.1 **This task is not required if you are upgrading from** Customized Web v7.5.2, v7.5.3 or v7.5.4	**66** Update all Copy controls on custom quick forms to reference a Summary view. The Copy control has been modified to only copy information from Summary views. If you have a custom quick form that contains a Copy control, you must change the quick form referenced in the LayoutFormName property of the Copy control to a Summary view. Use the following steps to change an existing quick form to a Summary view. **To change** 1 In Application Architect, verify that the quick form you want to edit is not open. The form must be closed to prevent overwriting your changes to the raw xml file with changes on the form. 2 In the VFS Explorer, open the quick form xml file referenced by the Copy control. If the project workspace is stored in the Virtual File System, use VFS Explorer. Otherwise, browse to it on the local disk under the project workspace folder. The form is located under: Model > Entity Model > *package* > *entity* > QuickForms > *quickformname*.main.quickform.xml. 3 Find the <Property name= "DefinitionType"…> node near the top of the file. For example: <Property name="DefinitionType" type="System.String"> Sage.Platform.QuickForms.QuickFormMainDetailViewDefinition, Sage.Platform.QuickForms, Version=7.2.1.1621, Culture=neutral, PublicKeyToken=null</Property> 4 Change the type value of the node to "Sage.Platform.QuickForms.QuickFormSummaryViewDefinition, Sage.Platform.QuickForms". For example: <Property name="DefinitionType" type="System.String"> Sage.Platform.QuickForms.QuickFormSummaryViewDefinition, Sage.Platform.QuickForms</Property> **Note** If the form you are editing has the "Version=…, Culture=neutral,…" information, you can remove it. 5 Save the file. 6 In the Project Explorer, right-click the project name containing the quick form, and then click **Reload Project**.
This task is required if you are upgrading from Customized Web v7.5.0 and v7.5.1 **This task is not required if you are upgrading from** Customized Web v7.5.2, v7.5.3 or v7.5.4	**67** Update consumption of SData feeds. If you have any code that consumes an SData feed, you must update the URL(s) and modify the way the payload is consumed. **To update the URL(s):** change "/dynamic" to "/dynamic/-" in your URL(s). **To update how your code consumes the payload:** determine how the payload format changed and update your code accordingly. **Note** You can write code to consume the feed directly or you can use the SData Core Client Libraries. They are available with documentation and examples at http://github.com/SageSalesLogix/SDataCSharpClientLib.

Required for	Task Description

This task is required if you are upgrading from Customized Web v7.5.0 and v7.5.1
 	
This task is not required if you are upgrading from Customized Web v7.5.2, v7.5.3 or v7.5.4
	68 Update how custom controls on AJAX panels register scripts.
	
	To ensure that custom controls work correctly with AJAX panels, change any script registration with the ScriptManager to the following approach.
	
	To register
	
	1 Make sure the control that is registering the script implements the IScriptControl interface, which is part of the ASP.NET AJAX framework. It requires you to define two methods, GetScriptDescriptors and GetScriptReferences. For example:
	
	```\npublic IEnumerable<ScriptDescriptor> GetScriptDescriptors()\nyield break;\n}\npublic IEnumerable<ScriptReference> GetScriptReferences()\n{\nyield return new\n ScriptReference("~/SmartParts/Opportunity/AddOpportunityProduct.js");\n}\n```
	In the previous example, we yield the script reference we want to add in the GetScriptReferences function. Add a yield statement for each ScriptReference needed.
	
	2 Register the control with the ScriptManager in the PreRender event of the control. For example:
	
	```\nprotected override void OnPreRender(EventArgs e)\n{\n if (DesignMode == false)\n if (ScriptManager.GetCurrent(Page) != null)\n(ScriptManager.GetCurrent(Page).RegisterScriptControl(this);\n}\n```
	3 Ensure that Sys.Application.notifyScriptsLoaded is called at the end of the referenced script, for example:
	
	```\n"if (typeof(Sys) !== 'undefined') Sys.Application.notifyScriptLoaded();\n```
	
	The script will now be correctly referenced when the control is added to the page during a partial post back.

This task is required if you are upgrading from Customized Web v7.5.0 and v7.5.1 **This task is not required if you are upgrading from** Customized Web v7.5.2, v7.5.3 or v7.5.4	**69** Update your Web customizations to point to the new mail merge DLLs. The following mail merge DLLs have been changed. The new mail merge DLLs will not conflict with any existing DLLs from 7.5.1 or earlier. Conflicts do occur with v7.5.2; therefore, a user cannot have both the ActiveX for v7.5.2 and v7.5.3 installed at the same time. • SLXDOCW.DLL replaces SLXDOC.DLL • SLXFAXW.DLL replaces SLXFAX.DLL • SLXMMENGINEW.DLL replaces SLXMMENGINE.DLL • SLXMMGUIW.DLL replaces SLXMMGUI.DLL • SLXWINFAXW.DLL replaces SLXWINFAX.DLL

Required for	Task Description
This task is required if you are upgrading from Customized Web v7.5.0 and v7.5.1 **This task is not required if you are upgrading from** Customized Web v7.5.2, v7.5.3 or v7.5.4	**Task 69 - continued** **Customizing the sage-mailmerge.js** The scripts MailMerge.vbs and MailMerge.js have been deprecated and can no longer be used. If you have customized MailMerge.vbs, MailMerge.js, or any script that calls into these scripts, your customizations must be replaced using the new Mail Merge API. This includes both VBScript CreateObject and JavaScript "new ActiveXObject" instantiation of the Mail Merge objects (e.g. SLXMMEngineW.MailMergeEngine). The functionality formerly in the scripts MailMerge.vbs and MailMerge.js has been refactored and incorporated into sage-mailmerge.js and sage-mailmerge-loader.js. The sage-mailmerge.js JavaScript implements the Sage.MailMergeService, Sage.SelectEmailInfo, Sage.CustomGroupExport, and Sage.ExcelExport JavaScript classes. The sage-mailmerge.js JavaScript, as well as its corresponding debug version (sage-mailmerge-debug.js), are generated files and should never be edited directly. These files can be regenerated using the third-party JS Builder application by opening and rebuilding the JS Builder Project file SageMailMerge.jsb, located in SlxClient\jscript. You can download the JS Builder application from http://code.google.com/p/js-builder/. The source files included in the SageMailMerge.jsb JS Builder Project are: • SlxClient\jscript\sage-mailmerge\ sage-mailmerge-service.js • SlxClient\jscript\sage-mailmerge\ sage-mailmerge-context.js **Important** Sage highly recommends that you do **not** edit these source files directly. If these files are edited directly any subsequent upgrade process will be more complex and may introduce JavaScript errors or defects. **To customize the JavaScript files** 1 Create a new JavaScript file named sage-mailmerge-overrides.js. 2 Add sage-mailmerge-overrides.js to the existing SageMailMerge.jsb JS Builder Project file and save the project. 3 In JS Builder, open the **Build Settings** tab. 4 In the **Combine compressed files to create specific output files** section, select **Sage Mail Merge**, and then click **Edit**. 5 In the **Output Target Properties** dialog box, drag the sage-mailmerge-overrides.js from the list of **Project Files** to the *bottom* of the list of **Included Files**. 6 Click **OK**. 7 Rebuild the SageMailMerge.jsb in JS Builder. When you rebuild the project the newly generated sage-mailmerge.js (and corresponding sage-mailmerge-debug.js) will include the appended content from sage-mailmerge-overrides.js at the end of these JavaScript files. When the Web browser loads the new sage-mailmerge.js JavaScript file, any overridden functions will automatically replace its original version.

Required for	Task Description
This task is required if you are upgrading from Customized Web v7.5.0 and v7.5.1 **This task is not required if you are upgrading from** Customized Web v7.5.2, v7.5.3 or v7.5.4	**70** Update your custom fax provider. As noted in the task 69, the SLXFaxW DLL replaces the SLXFax DLL. If you have a custom fax provider that runs on the Web, update the provider to use the IFaxProvider to interface with the SLXFaxW fax provider.
This task is required if you are upgrading from Customized Web v7.5.0 and v7.5.1 **This task is not required if you are upgrading from** Customized Web v7.5.2, v7.5.3 or v7.5.4	**71** Update custom forms that link to the Sage SalesLogix online help. If you have a custom form that links to a topic in the Sage SalesLogix Web Client help, you must update your form to use the PageLink control. Refer to the help topic "Linking a Help Topic to a Help Button" in the Application Architect online help for more information. **Note** You must use the PageLink control in a localized environment to link to the help file in the correct language folder based on your browser's language setting.
Optional	**72** Users will need to reset their user options when they log into the Sage SalesLogix Web Client. Instruct users to update their user options by clicking **Tools** > **Options**.
This task is required if you are upgrading Sage SalesLogix Mobile	**73** Convert your mobile customizations. 1 Change all modules (files) to use AMD loading. 2 Change all Ext calls to Dojo equivalents. 3 Verify that all modules names are fixed to reflect the new folder structure. 4 Verify that any property name changes do not affect your views or customization's. 5 Verify that the changed function names/removal do not affect your views or customization's. 6 Consider the newly added properties and methods to enhance existing customization's.

Required for	Task Description

This task is required if you are upgrading from Web v7.5.0
 	
This task is not required if you are upgrading from Web v7.5.1, v7.5.2, v7.5.3 or v7.5.4
	74 Upgrade Web Reporting.
	Note When upgrading from v7.5.1, v7.5.2, v7.5.3 or 7.5.4 your existing Web Reporting portal is updated when you build and deploy in task 76.
	
	Manual configuration of the Web Reporting Server in Application Architect is no longer required. Web Reporting is enabled by deploying the Web Reporting portal using the Deployment Explorer. The Web Reporting portal contains the Web Reporting application files and is deployed as a dependency of the SlxClient portal.
	
	If you are running Web Reporting at your main office and any Web Remote Offices, you must deploy the Web Reporting portal to both locations.
	
	To upgrade
	1 Sage SalesLogix DVD > **Standard Server Installation** > **Web Reporting Server**.
	   Note If the installation does not detect the necessary prerequisites, you will be prompted to install them. Click Install to allow Sage to install the required components or Cancel to stop the installation.
	2 In the **Sage SalesLogix Web Reporting** message box, click **Yes**.
	3 On the **Welcome** and **License Agreement** screens, read the information and accept the agreement, and then click **Next**.
	4 You will be prompted to enter some or all of the following information on the installation screens:
	   • **Create Web Site** – Select this check box if you want the installation to automatically create a Web Site.
	      • **Enter a Web site name** – Enter the name of the Web Reporting site in IIS. The default value is *SLX Web Reporting*.
	      • **Enter a Web Site Port** – Enter the port that the Web site will use. Note the port number for future reference. You must use port number 1025 or greater.
	      • **Enter a virtual directory name** – Enter the name of the virtual directory that will be set up for Web Reporting. The default virtual directory name is *slxwebrpt*.
	      • **Domain** – Enter the network domain where you created the WebDLL user.
	      • **User Account** – Enter the name of the WebDLL user. The default value is *WebDLL*. If you used a different name, type it exactly as you did when you created it.
	      • **Password** and **Confirm Password** – Type the WebDLL user's password.
	   • **Configure SalesLogix Database** - Clear this check box if you do not want to connect to the Sage SalesLogix database, but still want to create a Web site. When you clear this check box, the other options on this screen are disabled.
	      • **SalesLogix Server** – Select or enter the machine name where the Sage SalesLogix Web Server resides.
	      • **SalesLogix Database Alias** – Select or enter the name of the connection to the Sage SalesLogix database.
	      • **Search for servers on port** - Enter a port. By default, this is set to port *1706*.
	      • **User Name** – Enter a Sage SalesLogix Server user name (such as Admin).
	      • **Password** – Enter the password for the Sage SalesLogix user.
	5 After you have completed the information on the **SalesLogix Database Connection** screen, click **Next**.

Required for	Task Description
Upgrade Web Reporting	

This task is required if you are upgrading from Web v7.5.0
 	
This task is not required if you are upgrading from Web v7.5.1, v7.5.2, v7.5.3 or v7.5.4
	Task 74 - continued
	
	6 On the remaining screens, click **Install** and **Finish** to complete the installation.
	
	Note The installation process removes prior versions before installing the new applications. During this process, you may see the message "Assembly not Found." This is triggered between removing and replacing a service and does not indicate a problem with your installation. Click OK to continue the installation.
	
	7 The following steps are necessary when upgrading from v7.5.0. When upgrading from v7.5.1, v7.5.2, or v7.5.3, your existing Web Reporting portal is updated when you build and deploy in task 76
	
	8 Open the Application Architect Deployment Explorer.
	
	9 Add a new deployment.
	
	  a Right-click **Deployments**, and then click **Add New Deployment**.
	
	  b In the **Name** box, type Web Reporting
	
	  c In the **Description** box, type Web Reporting.
	
	  d Right-click **Deployment Targets** and then click **Add Target(s)**.
	
	  e In the **Select Target(s)** dialog box, select **IIS** and **slxwebrpt**, and then click **OK**.
	
	  f In the **Deployment Targets** tree view, expand IIS, and then select **slxwebrpt**.
	
	10 Set **IIS Target Settings**.
	
	  a In the **Server** box, type the name of the Web Reporting server.
	
	  b In the **Base Directory** box, type the Web Reporting installation path (for example, C:\Program Files\SalesLogix\Web Components).
	
	  c In the **Port** box, type the port used when installing Web Reporting (for example, 3334).
	
	  d In the **App Pool** box, verify the setting is **SalesLogix** (if Web Reporting is on the same server as the Web Host) or **SLX Web Reporting** (if Web Reporting is on a dedicated server).
	
	  e Select the **Restore Virtual Directory Settings on Update** check box.
	
	11 Click **Save**.
	
	12 Click **Deploy**.
	
	13 Open the SlxClient portal.
	
	  • For the main office: Double-click **Core Portals**, expand **Deployment Targets**, expand **IIS**, and then select **SlxClient**.
	
	  • For a Web Remote Office: Double-click **Remote Sales Client**, expand **Deployment Targets**, expand **Remote Office(s)**, and then select **SlxClient**.
	
	14 In the Advanced Options section, click **Web Dependencies**.
	
	15 In the **Manage Custom Service Host Entities** dialog box, click **Add**.
	
	The Service Host Entry Editor dialog box opens.
	
	16 In the **Entity Name** box, type *slxwebrpt*.
	
	17 In the **Service URL** box, type the URL for the Web Reporting portal. Use the format: http(s)://server_name:port_number/slxwebrpt.
	
	18 Click **OK**. Then, click **Done**.
	
	19 Save and deploy the Web site.
	
	20 Reset IIS.

Required for	Task Description
Build and Deploy your Web Portals	
This task is required if you are upgrading from Web v7.5.0, v7.5.1 or v7.5.2 **This task is not required if you are upgrading from** Web v7.5.3 or v7.5.4	**75** Add the SData portal to your Offline Web Client deployments. The SData portal must be added for Desktop Integration Module support. **To add** 1 In the Application Architect, open the **Deployment Explorer**. 2 Right-click the **Remote Sales Client** deployment, and then click **Edit Deployment**. 3 In the **Deployment Targets** tree view, right-click **Remote Users** and then click **Add Portal(s)**. 4 In the **Select Portal(s)** dialog box, select **sdata**, and then click **OK**. 5 Verify that the **Deploy Target** option and all Offline Web Client Remote Users in the **User Name** list are selected. 6 Click **Save**.
All Web	**76** Use the Application Architect to build and deploy the appropriate Web portal(s). Use the Rebuild Web Platform option on the Build menu when upgrading. See the "Rebuilding the Web Platform" and "Deploying Portals" topics in the Application Architect Help for instructions. **Note** Offline Web Clients must be deployed using the Virtual File System (VFS). Deployments created from a local file system are not supported and will not function correctly.
All Web	**77** Clear the web browser's cache. Before logging into the upgraded web client, instruct users to clear the cache of the web browser. For detailed steps, please refer to your browser's help.
All Web	**78** Run the Role Security utility to update roles and secured actions. **Note** If you modified the default roles by removing a role or removing a secured action from a role, note all your changes or create a backup of your roles before proceeding. The Role Security utility must be run to upgrade Web Client roles. The utility modifies existing roles and secured actions using the following rules: • If you removed a default Sage SalesLogix role, the utility adds the role. • If you removed a secured action from a default Sage SalesLogix role, the utility adds the secured action back to the role. • If you added a secured action to a default Sage SalesLogix role, the utility leaves the added actions in the role. • If you created custom roles, the utility does not modify custom roles.

Required for	Task Description

All Web	**Task 78 - continued**
	
	To run
	
	1 On the Administrative Workstation, browse to the …Program Files\SalesLogix folder.
	The utility is installed with the Administrative Tools and Servers installation.
	2 Right-click the **RoleSecurityInstall.bat** file, and then click **Edit**.
	3 Modify the following values to match your environment:
	  • SLXServer
	  • Alias
	  • Port
	  • Password
	4 Save your changes and close the .bat file.
	5 Double-click the **RoleSecurityInstall.bat** file to run the utility.
	Note For information on the arguments of this utility, type the utility name at a command line prompt in the SalesLogix install directory on the Administrative Workstation.

All Web on Windows 2003	**79** Update your SData configuration. **Note** For IIS7 and above, the Application Architect automatically configures the necessary portal settings. Sage SalesLogix now uses the .NET 4.0 framework so the aspnet_isapi handler mapping must be updated to reflect this change. **To update** 1 In IIS 6, expand your Web server machine, then expand **Web Sites**. 2 Expand the **SalesLogix** Web site (or the Web site where you deployed the SData portal), right-click the **SData** portal, then select **Properties**. 3 On the **Virtual Directory** tab, in the **Application Settings** section, click **Configuration**. The Application Configuration dialog box opens. 4 On the **Mappings** tab, under **Application Extensions**, select the **.ashx** extension, and then click **Edit**. The Add/Edit Application Extension Mapping dialog box opens. 5 Select the entire entry in the **Executable** box, right-click the **Executable** box, and then click **Copy**. 6 Select the **All verbs** option, and then click **OK**. 7 In the **Wildcard Application Maps** area, click **Insert**. 8 In the **Add/Edit Application Extension Mapping** dialog box, right-click the **Executable** box, and then click **Paste**. The file C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\aspnet_isapi.dll is pasted into the box. 9 Clear the **Verify that file exists** check box. 10 Click **OK**. 11 Click **OK** twice more. 12 Close IIS Manager.

Required for	Task Description
This task is required if you are upgrading from Web v7.5.0, v7.5.1 or v7.5.2 **This task is not required if you are upgrading from** Web v7.5.3 or v7.5.4	**80** Configure SData for Desktop Integration functionality. The Sage SalesLogix Desktop Integration Module is a collection of features to enhance your Sage SalesLogix Web experience. Desktop Integration requires SData. To use Desktop Integration functionality, you must: • Install and configure components on each Web Client computer. Web Client users configure the connection to the SData portal in the Desktop Manager. (Desktop Manager is installed with the Desktop Integration Module.) See task 86 for details.
This task is required if you are upgrading from Web v7.5.0, v7.5.1 or v7.5.2 **This task is not required if you are upgrading from** Web v7.5.3 or v7.5.4	**81** Set permissions to the Libraries folder for Desktop Integration functionality. The Sage SalesLogix Web site Libraries folder must have permissions set to enable Web Client users to download the Desktop Integration Module installation. **To set permissions on IIS 6** 1 On the Web Host, open IIS Manager. 2 Expand your server, expand Web Sites, expand your Sage SalesLogix Web site, and then expand the SlxClient folder. 3 Right-click the **Libraries** folder, and then click **Properties**. 4 Click the **Directory** tab. 5 In the **Execute permissions** drop-down list, select **Scripts only**. 6 Click **OK**. **To set permissions on IIS 7** 1 On the Web Host, open IIS. 2 Expand **Sites**, expand your Sage SalesLogix Web site, and then expand the SlxClient folder. 3 Click **Libraries**. 4 In the right pane, right-click **Handler Mappings**. 5 Click **Open Feature**. 6 In the right pane, click **Edit Feature Permissions**. 7 In the **Edit Feature Permissions** dialog box, select **Read and scripts** and clear **Execute**. 8 Click **OK**. 9 Reset IIS.
Customized Web	**82** Add your customizations to the MergeConfiguration.xml and PotentialMatchConfiguration.xml files. Use the customizations recorded in task 15 to merge your customizations with the updated Sage file. **To merge** 1 In the Web Client, convert a lead. Converting a lead automatically creates an updated MergeConfiguration.xml and PotentialMatchConfiguration.xml files. 2 In the Application Architect, open the Virtual File System Explorer. 3 In the tree view expand **Configuration**, and then expand **Global**. 4 Add your customizations to the updated **MergeConfiguration.xml** and **PotentialMatchConfiguration.xml** files.

Required for	Task Description
Remotes	**83** Cycle your Synchronization Server(s) so changes are sent to remotes. Use Custom Sync since it is not necessary to run subscription. **Note** See the "Creating a Custom Sync Cycle" topic in the Administrator Help for more information.
Web Remotes	**84** Instruct Offline Web Client users to log on to the Synchronization Client and sync immediately after upgrading. **Note** Ensure users enable "Apply Changes" during the synchronization cycle. **Offline Web Client Users and Web Reporting** Changes in this release allow Offline Web Client users to generate reports locally against their remote database and view the report in the new Sage SalesLogix Crystal Report Viewer (SLXCRViewer.exe). Offline Web Client users must have the "Use ActiveReporting" option enabled in the Web Client to use this feature. If Active Reporting is disabled, Offline Web Client users cannot run reports locally when disconnected. However, if they have a connection to the host Web Reporting Server, they can run reports that contain data from the host database. The following steps describe the upgrade to Active Reporting. 1 Offline Web Client users upgrade their Sage SalesLogix installation (task 46). 2 New remote portals containing Active Reporting components are built and deployed at the main office (task 76). The Remote portal should not include a Web dependency for Web Reporting Server. 3 The main office Sync Server cycles to prepare the portal for synchronization to Remotes (task 83). 4 Offline Web Client users run a sync cycle to receive the updated portal (task 84). 5 The Personal Web Server receives and updates the new portal. 6 In the Web Client > Options > General/Search tab, the Use Active Reporting option is enabled by default. 7 Offline Web Client users must download Desktop Integration to use Web Reporting (task 86). 8 Offline Web Client users can use reporting with data from their local machine.
All Web	**85** Import your upgraded project to the VFS. After upgrading, you may want to import your upgraded project (stored locally) to the Virtual File System (containing the older project). This ensures the project in the VFS is the most recent version. For detailed steps to import the project, see the "Importing from Another Project" topic in the Application Architect Help.

Required for	Task Description
All Web	**86** Instruct Web Client and Offline Web Client users to download the Desktop Integration Module. Desktop Integration extends Sage SalesLogix Web Client functionality by downloading client-side files for the following features: • Office Integration encompasses Mail Merge, Export to Excel, and Drag and Drop functionality. Administrators can also drag-and-drop library files. • Outlook Integration encompasses SendSLX and Record to History. Before users install and configure these features, ensure you configured SData as detailed in tasks 75 and 80. **Note** Offline Web Client users must download the Desktop Integration Module to use Web Reporting. **To install and configure** 1 Open the Web Client login screen. • For Web Client users, open the Sage SalesLogix Web site. • For Offline Web Client users, in your System Tray, right-click **Personal Web Server**, point to **SlxClient**, and click **Open Site**. 2 Click **Enhance SalesLogix**. You must be a Power User or Admin user to install Desktop Integration. 3 Click the associated **Find out more** link to open the "Installing and Using Sage SalesLogix Desktop Integration" help topic for browser specific configuration information.
	87 Understand the Intellisync for SalesLogix upgrade. **Note** If you did install 7.5.3 Hot Fix 6 prior to upgrading to 8.0, you do not need to reconfigure Intellisync after the upgrade. In this release the upgrade process updates Intellisync without additional installation or configuration steps. See the "Configuring Intellisync for SalesLogix" topic in the Sage SalesLogix Client help and Web Client help. **No Web Installation** If you do not have a Web installation (no Web Host or Application Architect), the service pack updates the portal on each main office and remote computer. • The main office portal is updated when you complete task 47. • Network Clients receive updates when you complete task 40.

Required for	Task Description
Upgrade Intellisync for SalesLogix	

Intellisync for SalesLogix	**Task 87 - continued**
	
	Web Installation
	
	If you have a Web installation, the service pack and build and deploy process updates the Intellisync portal. Ensure you select the SlxIntellisync portal when building and deploying.
	• The main office portal is updated when you complete task 76.
	• Network Clients receive updates when you complete task 40.
	• Remote Clients and Offices receive updates when you complete tasks 43 and 44.
	• Web Clients receive updates after you deploy the updated SlxIntellisync portal (task 76) and they run a synchronization cycle to receive it (task 84).
	
	After upgrading, each Client computer must run a synchronization cycle. When applying the service pack, the synchronization database still exists and you can synchronize normally. However, if you uninstalled and reinstalled using Add/Remove Programs, the synchronization database does not exist and you **MUST** use the **Re-Sync** option for your first synchronization cycle. If you do not use Re-Sync, you may see duplicates in your database.
	
	To run a Re-Sync, you must enable confirmations and then Re-Sync. Use the following steps only if you need to run a Re-Sync.
	
	To enable confirmations
	1 Open Intellisync for SalesLogix.
	2 Click **Setup**.
	3 On the **Connection Settings** screen, click **Application Setup**.
	4 Select **Appointments**, and then click **Settings**.
	5 Click **Advanced**.
	6 Click the **Appointments** tab.
	   a In the **Option** drop-down list, select **Confirmations**.
	   b Verify the **Confirm changes and additions (Recommended)** option is selected.
	   c Click **OK**.
	7 Click **OK**.
	8 Repeat steps 4 through 7 for Contacts and ToDo's.
	9 Close the **Setup** dialog box.
	10 On the **Connection Settings** screen, click **OK**.
	
	To Re-Sync
	1 With Intellisync open, click **Sync Now**.
	2 On the **Confirm Edits** screen, click **Re-Sync**.

Required for	Task Description
Perform Post-Upgrade Tasks	
This task is required if you are upgrading from Web v7.5.0, v7.5.1 or v7.5.2 **This task is not required if you are upgrading from** Web v7.5.3 or v7.5.4	**88** Add all existing Web users to the Standard Role. By default, users created in the Web Client are assigned the Standard User role. This role allows add, edit, and delete permissions to Web Client non-administrative entities such as accounts, contacts, and opportunities. Users created in the Administrator are not assigned a role and role security cannot be set in the Administrator. If you have existing users or create new users in the Administrator, you must assign users to the Standard User role. **To assign** 1 Open the **Users List** view in the Web Client. 2 Select the users that should have add, edit, and delete permissions to non-administrative entities in the Web Client. 3 In the User Tasks pane, click **Add to Role**. 4 In the **Select Role** dialog box, lookup and select the **Standard User** role, and click **OK**. 5 Click **OK**.
This task is required if you are upgrading from Web v7.5.0, v7.5.1 or v7.5.2 **This task is not required if you are upgrading from** Web v7.5.3 or v7.5.4	**89** If necessary, enable Windows Authentication in browser. The following instructions configure Windows Authentication for Web Client users accessing Sage SalesLogix. Use one of the following procedures as appropriate for your browser type. **To enable on Internet Explorer** 1 On each Web Client computer, open your browser. 2 On the Tools menu, click Internet Options. 3 Click the Security tab, select Trusted Sites, and then click Sites. 4 In the Add this Web site to the zone: text box, enter either the Web Server name or the IP address to the Web Server (http://10.40.0.35). **Note** For users using Windows Authentication, the URL address must include the windows.aspx page (http://10.40.0.35/slxclient/windows.aspx). **To enable on Firefox** 1 On each Web Client computer, open your browser. 2 In the Location Bar, type about:config, and then press Enter. 3 If a warranty message opens, click I'll be careful, I promise!. 4 Double-click the network.automatic-ntlm-auth.trusted-uris preference. 5 In the Enter string value box, type your Web Host URL using the format http://hostserver:port. 6 Click OK.

Required for	Task Description
Optional	**90** Enable HTTP Compression on IIS 7 with IIS 6 compatibility. **Note** IIS 6 compression is documented in the *Sage SalesLogix Implementation Guide*. If your Web site(s) use large amounts of bandwidth, or if you would like to more effectively use bandwidth, you may want to consider enabling HTTP compression. HTTP compression provides faster transmission time between compression-enabled browsers and IIS. Compression has no one-size-fits-all solution that enables all users to achieve the same results in different environments. However, there are some guidelines that may increase performance in terms of lighter network bandwidth at the expense of CPU and memory used by the IIS server. You must be a member of the Administrators group on the local computer to enable compression. By default, only static compression is enabled in IIS 7. Enabling dynamic compression will provide significantly smaller files over the network. In addition, there are a number of configuration elements that may be modified to provide more benefits. You may need to try different settings to achieve optimum results. The following information can be used as a guide to implement compression in your Sage SalesLogix implementation. To help evaluate and test your specific compression settings, you may want to use a third-party HTTP compression debugging tool. **To enable** 1 Open IIS Manager, and select your Sage SalesLogix Web site. 2 Under **IIS**, double-click **Compression**. 3 Select the **Enable dynamic content compression** check box. If necessary, enable the dynamic content compression module in Server Manager (Roles) to enable the check box. 4 In the **Actions** pane, click **Apply**. 5 If necessary, modify the **ApplicationHost.config** file using the following example. This file is stored in %SystemRoot%\System32\inetsrv\config\ApplicationHost.config. In IIS 7 the configuration file replaces the Metabase Explorer compression values used in IIS 6. For more information, see the Microsoft IIS Web site at http://www.iis.net/ConfigReference.

Required for	Task Description
Optional	**Task 90 - continued** • Example Configuration Settings: ```\n<httpCompression directory="%SystemDrive%\inetpub\temp\IIS Temporary\n Compressed Files" maxDiskSpaceUsage="500" noCompressionForHttp10="false"\n noCompressionForProxies="false" >\n <scheme name="gzip" dll="%Windir%\system32\inetsrv\gzip.dll"\n dynamicCompressionLevel="10" staticCompressionLevel="10"/>\n <scheme name="deflate" dll="%Windir%\system32\inetsrv\gzip.dll"\n dynamicCompressionLevel="10" staticCompressionLevel="10" />\n <staticTypes>\n <add mimeType="text/*" enabled="true" />\n <add mimeType="message/*" enabled="true" />\n <add mimeType="application/*" enabled="true" />\n <add mimeType="*/*" enabled="false" />\n </staticTypes>\n <dynamicTypes>\n <add mimeType="text/*" enabled="true" />\n <add mimeType="message/*" enabled="true" />\n <add mimeType="application/*" enabled="true" />\n <add mimeType="*/*" enabled="false" />\n </dynamicTypes>\n</httpCompression>\n```
This task is required if you are upgrading from v7.5.0 and v7.5.1 **This task is not required if you are upgrading from** v7.5.2, v7.5.3 or v7.5.4	91 Enable UTF-8 support for Internet Protocols (Microsoft Office 2007). If users export extended characters from Sage SalesLogix to e-mail, UTF-8 support must be enabled. Enable UTF-8 support in Outlook from Tools > Options > Mail Format tab > International Options > Internet Protocols section.

Required for	Task Description

This task is required if you are upgrading from v7.5.0 and v7.5.1
 	
This task is not required if you are upgrading from v7.5.2, v7.5.3 or v7.5.4
	92 If necessary, create a 32-bit OLE DB UDL file for your 64-bit system.
	
	If you are running Sage SalesLogix on a 64-bit system, and you have installed the provider extensions, you must run the Data Link Manager in 32-bit mode. Use the following steps to configure the Data Link Manager for a 64-bit system.
	
	Both 32-bit and 64-bit machines must run the Data Link Manager and access the SLX OLE DB Provider in 32-bit mode.
	
	To create
	1 Using Notepad, create a blank UDL file named "test.udl".
	2 Save the UDL file to your C: drive.
	3 Open the blank UDL file doing one of the following:
	   • For a 64-bit machine, run the following command: C:\Windows\syswow64\rundll32.exe "C:\Program Files (x86)\Common Files\System\Ole DB\oledb32.dll",OpenDSLFile C:\test.udl
	   • For a 32-bit machine, double-click the file in the file manager and run C:\Windows\system32\rundll32.exe "C:\Program Files\Common Files\System\OLE DB\oledb32.dll",OpenDSLFile <SalesLogix install folder>\'test'.udl
	4 Create a new database connection and make sure you select the SalesLogix OLE DB Provider on the Provider tab.
	5 Save the connection.
	6 Move the UDL file to all computers that require a connection to Sage SalesLogix.
	
	By default, the file should be stored on the C drive.

Required for	Task Description
This task is required if you are upgrading from a Unicode enabled installation **This task is not required if you are upgrading from** a non-Unicode installation	**93** If your Sage SalesLogix installation is Unicode enabled, manually update the Provider Extensions. In some Unicode installations, users with multi-byte character user names cannot log on to Sage SalesLogix. To fix this issue, complete the following configuration steps. **To configure** 1　On the Sage SalesLogix media, browse to the SDK Examples > Provider Extensions folder. 2　Copy the Provider Extensions folder to your Sage SalesLogix Server (or to any computer that can connect to the Sage SalesLogix Server). 3　From the Provider Extensions folder, open the test.udl file in the OLE DB Core Services application. 4　Set the following values in the test.udl file, test the connection, and then click **OK**. 　• Sage SalesLogix Server 　• Database Name 　• Sage SalesLogix user name (generally admin) 　• Sage SalesLogix user name password 　• Select the Allow Saving Password option 5　Open a command prompt and change the directory to the Provider Extensions folder. 6　At the command prompt run the following command: slxproviderplugin -m slxproviderextension\provider_extensions.xml -u test.udl -i -d The "** SUCCESS: Plugin installed successfully" message displays when the plugin has been installed. 7　Restart all Sage SalesLogix Servers.
All Web	**94** Instruct Web Client users to set the Scrollbar Item setting to 17 or less. The Scrollbar Item setting must be 17 or less on each Web Client computer or scrollbars may not display correctly. Verify the setting from the Control Panel. 　• (XP/2000/2003) The Scrollbar option is set in the Control Panel > Appearance and Themes > Display properties Appearance tab > Advanced > Item drop-down menu. 　• (Vista/7/2008) The Scrollbar option is set in the Control Panel > Personalization > Window Color > Window Color and Appearance > Item drop-down menu.

You have completed your upgrade. See the " in this Release" topic in the online Help systems for an explanation of the new features.

Chapter 2
Changes in this Release

Version 8.0 is a cumulative release which includes all changes released since version 7.5.0.

> Any change in the underlying architecture in the base product causing customizations to not function properly is identified as a [Breaking Change].

See the "What's New in this Release" topic in the online Help systems for an explanation of the new features.

Changes in Version 8.0

This version contains defect fixes and the following updates.

General Enhancements

- Ability to set default Mail Merge templates for leads.
- Additional standard SpeadSearch indexes.
- Ability to secure a SpeedSearch index to limit the users that can retrieve data from that index during a SpeedSearch.
- New large icons toolbar in the Sage SalesLogix Client.
- More applications and features support Unicode. For example, the Network Client can now read and write Unicode.
- A new Location field on the Activity, History, and Event views.
- Windows 7 support for thumbnail preview images.
- [Breaking Change] The request validation feature in ASP.NET 4.0 is enabled for all requests. This provides protection against cross-site scripting attacks.

Web Client Enhancements

- Ability to lock out Web Client users after failed login attempts and reset locked out users.
- New look and feel of the Web Client with updated styles and colors.
- List view groups remember your sort and column width changes.
- Perform a Lookup from the left Navigation bar in order to look up any type of record.
- New Alerts feature tells you when there is an activity alarm or when you have a new unconfirmed activity notification. You can set Alerts options to customize your alerts.
- Improved Activities:
  - The Availability tab allows you to view other user's calendars and determine availability for other user's and resources.
  - A Location box allows you to enter a location for a phone call or meeting or automatically add the location if one is selected from the Availability tab.
  - Activity confirmations now provide more information to help you determine whether to accept or decline an invitation.
  - Updated Activity List view has new tabs, improved filters and Common Tasks. Use the My Activities tab to view activities where you are the leader or a member.

- You can view contact and account or lead information, like a phone number, by hovering over the name in the activity detail view or Alerts view.
- New and updated activity and calendar user options:
  - New activity user options allow you to set defaults per activity type.
  - New calendar user options allow you to determine what activity information to show on the calendar, the days of your work week, and your first day of the week. You can also set an option to remember the users whose calendars you selected to view.
  - Options are now shared between the Sage SalesLogix Client and Sage SalesLogix Web Client. Some users may need to reset their options after upgrading to the latest version.
- Calendar has been updated and includes new features such as:
  - Ability to customize which days comprise your work week.
  - Ability to see up to three users on each calendar view.
- Accounting Integration now includes X3 integration.
- Move Contact and Duplicate Contact are now handled separately. The new methods allow you to determine what information is moved or reassigned.
- New Mail Merge templates for leads.
- Ability to share Mail Merge templates with other users.
- The Opportunity menu has been removed. Update Multiple Opportunities and Opportunity Statistics are now available on the Opportunity List view right-click menu.
- Features for Administrators and those with administrative rights:
  - The Standard Role now grants view access to products and packages, instead of add, edit, or delete access.
  - The Security Manager allows you to manage field-level access within security profiles.
  - Resource management allows you to identify if a resource is a location.
  - The Web Form Designer enables you to make simple changes to a view. For example, you can change captions, move things around, or add and hide information. For more information, refer to the "Form Designer - Working with Forms" topic in the Web Client help.
- Now fully supports Unicode.

Application Architect Enhancements

General

- You can quickly find and open project items without expanding the Project Explorer or Deployment Explorer using the new Find Item command or Ctrl-T. Refer to the "Navigating and Opening Project Items" topic in Application Architect help for more information.
- The Select Project Backup File dialog box has been enhanced to include a Backup Date column and re-sizable columns.
- You can now bundle deletes of support files.
- To improve bundling performance, you can now select the portals that should be examined for differences. Refer to the "Creating a Manifest by Project Differences" topic in the Application Architect help for more information.
- Application Architect is configured to use the Sage Cache server to improve performance of the VFS.
- In general caching, cookies are no longer used. Client-side caching uses local storage for things such as pick lists and session storage for things such as user options. Only session ID and remembered login use cookies now.

- Use the Sage Job Server to schedule jobs. It enables you to execute tasks out of process, freeing up worker threads and memory in the Web Host. Good candidates for this are resource intensive tasks and long running tasks. Refer to the "About the Sage Job Server" topic in the Application Architect help for more information.

- Error logging has been enhanced with additional detail including user and system information for better debugging. Also, you can configure aspects of error handling such as the level of detail provided per user. Refer to comments in the appSettings.config file of the Sage SalesLogix portal for more information.

- Client-side form validation has been added to the Web Client. Refer to "About Client-Side Form Validation" in the Application Architect help for more information.

- An image sprite is used to reduce the number of server requests for icons. Customizations can also use the global image sprite to reduce server requests. Refer to the "Using the Global CSS Image Sprite" topic in the Application Architect help for more information.

- The Move Contact process provides several opportunities for customization. Refer to the "Customizing the Move Contact Process" topic in the Application Architect help for more information.

- The OLE DB provider extension for history security has been de-coupled from the activities security extension so it can be enabled and disabled independently. Now it is enabled for all LAN, Web, and Mobile clients by default so history records are secured. Refer to the "OLE DB Provider Extensions" topic in the Application Architect help for more information.

- You can add MIME types for attachments. Refer to the "Adding File Types for Attachments" topic in the Application Architect help for more information.

- The work practice survey feature was removed.

- [Breaking Change] The task pane workspace update panel has been removed. If you have custom tasklets that reside in the task pane workspace and have server-side controls that rely on postback behavior, ensure they have their own update panels.

Quick Forms

- Dynamic customization - You can make changes to quick forms in Application Architect and see them real-time in the Web Client. Allows developers to rapidly lay out quick forms without repeated builds and deploys. Refer to the "About Dynamic Customization" topic in the Application Architect help for more information.

- The style of some controls has changed. For example, single-select pick lists use a down arrow while multi-select pick lists still have an ellipsis. Refer to the control topics in the Application Architect help for more information.

- C# Code Snippets are no longer labeled Obsolete. Because the code is embedded in the forms, the snippets are compatible with dynamic customization; when you make changes to the code, it will update live on the web site. Refer to the "About Dynamic Customization" topic in the Application Architect help for more information.

- A quick form editing capability was added to the Sage SalesLogix Web Client. For information on how the Form Designer in the Web Client interacts with the Quick Form Designer in Application Architect, refer to the "Editing a Quick Form in the Web Client" topic in the Application Architect help.

- The Quick Form Designer has a new look and work flow. Refer to the "Editing a Quick Form" topic in the Application Architect help for information on the Quick Form Designer:

  - Toolbar Editor - A graphical toolbar editor on the design surface lets you easily create standard Save and Delete buttons as well as business rule buttons and control buttons. Relocation of buttons between left, middle, and right is done through drag and drop instead of delete and recreate. Refer to the "Toolbar Editor" topic in the Application Architect help for more information.

  - Control layout pane - The control layout pane features a reorganized context menu. Related controls are grouped on submenus and row and column commands are

collected together. Refer to the "Editing the Control Layout" topic in the Application Architect help for more information.

- Data source/non-visual control well - The Non-Visual Controls area has a constant presence at the bottom.

- Resource editor - The Resource Editor is accessible from the design surface and has been changed to improve work flow. Refer to the "Working with Resources" topic in the Application Architect help for more information.

- Action Designer/Code Editor - A code tab lets you manage all actions and JavaScript properties from one place. It introduces a rich JavaScript editing experience. Refer to the "Editing Actions and JavaScript Properties" and "JavaScript Editor" topics in the Application Architect help for more information.

- Custom controls can be added to the Quick Form Designer insert menus. Refer to the "Hosting Custom Controls in the Quick Form Designer" topic in the Application Architect help for more information.

- Add New Property command - Allows you to quickly create a new dynamic entity property and a bound control with the Quick Form Designer. This eliminates the need to add a column to the database before using it on a quick form. Refer to the "Inserting a New Entity Property" topic in the Application Architect help for more information.

- Swap Controls command - Allows you to trade the position of two controls. Refer to "Move Controls" in the "Editing the Control Layout" topic in the Application Architect help for more information.

- The ability to host Silverlight controls has been added to quick forms. Refer to the "Silverlight Control Host" topic in the Application Architect help for more information.

- A Client Script Include property was added to reduce the need for master page manipulation. Refer to the "Including Client Scripts in Smart Parts" topic in the Application Architect help for more information.

- A Client Resource Keys form property was added to give the ability to make strings used in JavaScript localizable. Refer to the "Form Properties" topic in the Application Architect help for more information.

- An Extended Properties property was added to forms and controls. It allows you to define key/value pairs to pass into the smart part code generation process. Refer to the "Extending Control and Form Properties" topic in the Application Architect help for more information.

- A Sort Direction property was added to columns of grid controls for specification of default sort order. Refer to the "QFDataGridCol Collection Editor" topic in the Application Architect help for more information.

- Prefilter properties can be made visible in the Lookup By section of a lookup dialog box where users can then modify and remove them. Refer to the "LookupPreFilter Collection Editor" topic in the Application Architect help for more information.

- The Required property was added to more controls: Currency, DateTime Picker, Dependency Lookup, Numeric, Person Name, Radio Group, and Time Zone. Refer to the "Required" topic in the Application Architect help for more information.

- The Lookup, Owner, and User controls use SData to query the database which requires that the lookup entity have its SData feed turned on. Refer to the "Generating SData Feeds" topic in the Application Architect help for more Information.

- Editable Grid Control was enhanced:

  - The Relate button of the Editable Grid control has a new property, Conditional Show, which allows you to specify a condition and message for when a message should be displayed instead of the lookup dialog box. Refer to Conditional Show in the "EditableGridToolDefinition Collection Editor" topic in the Application Architect help for more information.

  - Preview Property was added to the Editable Grid control so you can configure a preview pane for the grid. Refer to the "Preview Property" topic in the Application Architect help for more information.

- Column types (Date/Time, Email, Phone, SLX User, and Pick List Formatter) and properties (Abbreviation Length, Custom Format Function, Sort Direction) were added to the Editable Grid control. Refer to the "QFDataGridCol Collection Editor" topic in the Application Architect help for more information.

- The IsEditable property was removed from the SLXUser column type.

- The default value of column Width=-1 was changed from auto to 10 em. . Refer to the "QFDataGridCol Collection Editor" topic in the Application Architect help for more information.

- The ability to specify where toolbar icons merge with those on the smart part toolbar and the ability to hide icons in Insert rendering mode were added. For more information, refer to Merge Control Id, Merge Position, and Display In Insert properties in the "EditableGridToolDefinition Collection Editor" topic in the Application Architect help.

- The Aggregate control displays as two buttons instead of a link. Refer to the "Aggregate Control" topic in the Application Architect help for more information.

- Removed Generate Member property from Web Browser, Aggregate, and Horizontal Separator controls.

- Deprecated Display Exchange Rate Code property in Currency control and columns. If the value is set, it is ignored. Now the exchange rate code always displays when multi-currency is enabled.

- Deprecated As Hyperlink option for Display Mode property of Currency control.

- Removed Multiselect property from Owner control.

- Detail views built with quick forms now implement a splitter bar between the main content and tab workspace which enables users to view both at once when space is limited.

- The Numeric control now does its formatting client-side but is not compatible with Data Type=Double together with Is Percentage=True. Refer to the "Numeric Control" topic in the Application Architect help for more information.

- The Popup Grid Columns collection of an Aggregate control now allows for the currency column type.

- String lookups can be configured to append results instead of replacing the existing string. Refer to the "Append String Values" topic in the Application Architect help for more information.

Portal Manager

- The user interface has been streamlined for the different purposes of the four portal types: UI Portal, Service Portal, ASP.NET Portal, and Mobile Portal. Refer to the "Using the Portal Manager" topic in the Application Architect help for more information.

- You can customize the Sage Mobile portal by adding custom modules. Refer to the "Adding a Custom Module to a Mobile Deployment" topic in the Application Architect help for more information.

- Portal types Mobile Portal Template and ASP.NET Portal Template were added to the New Portal Wizard. Refer to the "Using the New Portal Wizard" topic in the Application Architect help for more information.

- Portal - Advanced tab was renamed to the Services tab. The new Scope property allows you to set the scope of the service to Session or Application. Refer to the "Services Settings" topic in the Application Architect help for more information.

- Portal - The Group Tab Context Menu can be customized for list views. Refer to the "Adding a Context Menu Item" topic in the Application Architect help for more information.

- Portal - Context menus now have Applied Security property to get or set the name of a secured action. Refer to the "Navigation Group and Item Settings" topic in the Application Architect help for more information.

- Page - Two configurable properties were added to the Portal Page Advanced tab: Applied Security and Script Include. Refer to the "Configuring Application ASP.NET Page Properties" topic in the Application Architect help for more information.
- Page – The Base Page Type and Entity Type configurable properties and the Configure Base button have been moved from the Portal Page Advanced tab to the Page Information area.
- The new Sage Job Service allows long running processes to run from a file system instead of through the web host. This frees worker threads in the web host and allows the processes to fully run without the http request timing out. The Sage Job Service also allows for scheduling processes. For more information, refer to the "Understanding the Job Service Portal" topic in the Application Architect help.
- The Cache Service is an implementation of ICacheProvider that is available in the Services Collection. The ICacheProvider service supports in memory and out of process (memcached) providers. Out of process cached data survives process restarts and improves performance.

Deployments

- Sage SalesLogix Web Portals now take advantage of the 4.0 .NET Framework.
- The Target Name property in deployment settings was added to identify deployment targets if you have multiple deployment targets of the same type.
- Application settings were externalized from the web.config file into an appSettings.config file. Refer to the "Editing Configuration Files" topic in the Application Architect help for more information.
- Deployment from the VFS-based Application Architect project is required for remotes and recommended for production deployments. Deployment from a local file system-based project is still a good option for development. Refer to the "Choosing a Project Storage Location - Virtual and Local File Systems" topic in the Application Architect help for more information.
- When deploying the SData portal to IIS 7 on Windows 2008, the portal no longer needs additional configuration in the IIS Manager such as handler mappings.
- Added a Cancel Deployment button to the Application Architect toolbar and moved the deployment messages to the output window.
- The Sage Job Server portal was added to the Core Portals. For upgrades, this must be added manually. Refer to the "Configuring Deployment Settings" topic in the Application Architect help for more information.

Entity Model

- Filter changes no longer require portal deployment. Filter information in the source project workspace/model is accessed directly by the portal. Refer to the "Deploying a Filter" topic in the Application Architect help for more information.
- Dynamic Entities, Properties, and Relationships can be created so they do not require a platform build. Refer to the "Understanding Static and Dynamic" topic in the Application Architect help for more information.
- Entity properties in the Web Client have been set to be audited to match Record to History settings in the Sage SalesLogix Network Client.
- Application Architect upgrades projects so that Manage Schema=True for entities in the SalesLogix Activity Support and SalesLogix Application Entities packages to enable dynamic customization of these entities. This change also enables the addition of properties to entities in these packages with Application Architect commands such as Modify Entity and Insert New Entity Property.
- A new entity data type of Url was added. When you drop an entity property of data type Url onto a form, a URL control is created.

- The NHibernate second level caching option was exposed on the Entity properties tab for potential performance improvements. Refer to Cache Usage in the "Entity Properties Window" topic in the Application Architect help for more information.
- Removed the legacy Application Support package.
- Query builder display names now come from the localizable entity and property names in the project model instead of fixed resources in the database.
- [Breaking Change] Date filter special values of todaystart, todayend, tomorrowstart, tomorrowend, yesterdaystart, and yesterdayend were added. Refer to the "Adding Custom Filters" topic in the Application Architect help for more information.

Security

- You can define new secured actions when you are configuring portal pages, navigation groups, menus, and context menus. Refer to the "Creating a Secured Action" topic in the Application Architect Help for more information.
- You can now override the default security implementation in the Sage SalesLogix OLEDB provider for individual users. Refer to the "Customizing Sage SalesLogix Security" topic in the Application Architect Help for more information.

Builds

- Build output and search path locations can be directly opened in Windows Explorer. Refer to the "Changing Build Output or Search Path Locations" topic in the Application Architect Help for more information.
- Clean Build Folders command added to Build menu. Refer to the "Cleaning Build Folders" topic in the Application Architect Help for more information.

Mashups

- Mashup changes no longer require portal deployment. The new compliant SData adapter uses the source project workspace/model. Refer to the "Deploying a Mashup" topic in the Application Architect Help for more information.
- The Query processor was renamed to HQL processor.
- A new processor type was added. The SQL processor executes a query on local or remote databases. Refer to the "SQL Processor Type" topic in the Application Architect help for more information.
- HQL/SQL Syntax Highlighted Editor - A new syntax highlighted editor has been created to make editing SQL and HQL a little more pleasant. It can be launched from the ellipsis on the respective Query properties in the property grid.
- Authenticated Feeds - User Name and Password properties have been added to the Feed processor so that feeds requiring authentication can be consumed. These properties behave in exactly the same way as those on the SData processor. They support inline expressions evaluated at runtime. Refer to the "Feed Processor Type" topic in the Application Architect help for more information.
- Most string properties support inline expressions evaluated at runtime. Refer to the "Inline expressions" topic in the Application Architect help for more information.
- A new help topic, "Creating a Custom Mashup Processor", has been added.

Analytics

- Widgets are HTML5 compliant.
- Pie charts can be configured for any number of slices. Also, you can set the maximum number of items to appear in bar, column, and line charts that are based on distinct filters.

- "Understanding Dashboard Widgets" help topic was enhanced to explain how to add Entities, Dimensions, and Metrics used in the charting editors. How to control the order of items in a funnel chart was included.

SData

- SData 1.1 specification compliance: support for the language query argument was added, bringing SalesLogix SData to full implementation of section 3.14 of the specification. Refer to the "Sage SalesLogix Implementation of Sage Data (SData) 1.1" topic in the Application Architect Help for more information.
- Additional functions are supported in SData queries. Refer to the "SData - Supported Functions" topic in the Application Architect Help for more information.
- The library of SData endpoints continues to be expanded. For more information, refer to the white paper, *SData Endpoints*.

API

- Added methods for easier ad hoc SQL method calling. Usually you should go through the entity model, but there are times when it is more beneficial to go through SQL. See methods MySlx.Data.ExecuteSql and MySlx.Data.ExecuteWithReturn.
- The Dependency Lookup control now supports dynamic changes to filters. In code, clear existing filters, ensure the cache is cleared using the InvalidControlCache method, and add new filters.

Documentation

- The following new help topics were added:
  - Adding a Custom Entity to a Merge Business Rule
  - Comparing Nulls in SData queries
  - Creating a Custom Mashup Processor
  - Loading Multiple Versions of jQuery
- "Understanding Dashboard Widgets" topic was enhanced to explain how to add Entities, Dimensions, and Metrics used in the charting editors. How to control the order of items in a funnel chart was included.
- "About Filters" topic was enhanced.

Sage SalesLogix Mobile

- Added an editable picklist for Activity Duration that allows you to enter duration with autocomplete.
- Added Activity Location property to Activity entity to match Web Client.
- Activity User Options are used for defaults on alarm, duration, timeless and auto-rollover settings.
- Application Level Default Values – support for default values in customizations at application level as well as entity level.
- Customizations and upgrades are easier to deploy.
- Contextual Action Menu – quick actions available in list views based on entity and availability of data.
- Added the ability to create an activity for contact, opportunity or ticket without first selecting account.
- A configurable number of errors are saved to local storage and can be viewed under Settings.
- Addresses added to related items under Account and Contact views allowing users to view all related addresses.

- Address templates can be applied to detail views based on region.
- Opportunity Contacts can be added or removed from an Opportunity on mobile device.
- Recurring Activities can be scheduled and edited from mobile device.
- Sage SalesLogix Mobile uses the same role-based security as the Web Client.
- Views adjust to landscape views to optimize usability.
- Events can be scheduled from mobile device and are displayed on the Calendar.
- Ticket Activities can be created/displayed/edited as a related item for a Ticket.
- Toolbar Customizations - RegisterCustomization API facilitates adding toolbar items.
- View Address Quick Action opens Google Maps on a separate tab in desktop browser.

Changes in Version 7.5.4

In addition to improved product stability due to defect fixes, v7.5.4 contains the following new functionality.

Deprecated Components

- Removed the Sales Potential field from the Web Client Insert Opportunity view. Users can edit the Sales Potential using the Opportunity Snapshot after the opportunity is saved.
- Removed support for Empulse (Marketing Services), Web Phones, and Legacy Web (Web versions earlier than 7.2). Web Phones and Legacy Web support were deprecated in v7.5.2.

Accounting Integration

This release adds the ability to integrate Sage SalesLogix with any accounting system that supports Sage CRM ERP Contract. Accounting integration is managed using the Web Client. The Network Client does not support accounting integration. See Chapter 4, "Configuring Accounting Integration" to enable and configure accounting integration.

General Enhancements

- Added the ability to customize Desktop Manager and repackage it into the installer. For instructions, refer to the Readme.txt file located on the Sage SalesLogix media in the SDK Examples\DesktopManager folder.
- Added the ability to specify the Desktop Integration installation folder. This allows you to install to a folder such as All Users which allows any user to have Desktop Integration load automatically on startup.

  To install Desktop Integration to a different folder, log on as a user with administrative rights and run the following command line: SlxDesktopIntegrationSetup.exe /s /v"/qn INSTALLDIR=\"C:\Program Files\<*installationfolder*>\""

  Change "C:\Program Files\<*installationfolder*>\" to your specific installation path.
- Updated the Administrative Tools and Servers, Offline Web Client, and Remote Client installations for Microsoft SQL Server Express 2008 R2. The Sage SalesLogix installations will set the following:

Scenario	Action
Microsoft SQL Server Express is not detected.	Install Microsoft SQL Server Express 2008 R2 and a SalesLogix instance.

Scenario	Action
A previous version of Microsoft SQL Server Express without a SalesLogix instance is detected.	Install Microsoft SQL Server Express 2008 R2 and a SalesLogix instance.
A previous version of Microsoft SQL Server Express with a SalesLogix instance is detected.	Upgrade only the SalesLogix instance to Microsoft SQL Server Express 2008 R2.

- Added the SalesLogix Cache Server for performance enhancements. The SalesLogix Cache Server stores data on a separate server so the cache is real-time. The service (Sage.SalesLogix.CacheService.exe) runs the executable that handles the caching. The web.config and hibernate.xml files have been updated for the new caching support. Any customizations to these files must be merged.

- [Breaking Change] Updated the Account entity for performance improvements. The following changes have been made to the Account entity:

  - Removed the following calculated properties: NumberOfOpenTickets, NumberOfTotalTickets, OpportunitiesClosedLostCount, OpportunitiesClosedLostTotal, OpportunitiesClosedWonCount, OpportunitiesClosedWonTotal, OpportunitiesOpenCount, OpportunitiesOpenTotal, OpportunitiesInactiveCount, OpportunitiesInactiveTotal.

  - Added a new Get Method called GetTicketStats();

  - Modified the AccountOpportunites quick form to bind to DataSource dsOppStats By Method GetOpportunityStats() that returns a component view with property names of String[] propNames = {"OpenCount", "OpenTotal", "ClosedWonCount", "ClosedWonTotal", "ClosedLostCount", "ClosedLostTotal", "InactiveCount", "InactiveTotal"} that binds to the QuickForm controls.

  - Modified a quick form under Ticket named AccountServiceInfo which sets the Total Ticket Count and Open Ticket Count on the Load method of the form by using the Ticket.Account.GetTicketStats() Method and then setting the control explicitly from the component view with property names of String[] propNames = {"OpenTicketCount", "TotalTicketCount"}

- Added Sage SalesLogix Mobile Classic Blackberry 6.0 compatibility.

- The SData Dynamic Adaptor is now configured in the Web Client. When writing Web Client customizations, this allows you to consume SData from the Web Client using the SLXData.ashx endpoint.

- Basic Authentication is now the default setting in the SData portal configuration. Basic Authentication does not require the WebDLL user to be associated to a Sage SalesLogix user.

- Added a configuration option for Windows Authentication to restrict Web Client users from logging on if the user is not found in Sage SalesLogix. The current configuration allows the user to log on manually if the user is not found in Sage SalesLogix. See the "Enabling Windows Authentication Access for Web Client Users" topic in the Administrator Help for instructions.

- Added the Sage SalesLogix Component Finder utility. This utility compiles a list of the Sage SalesLogix components and their version numbers that are installed on the machine where the utility is run.

Web Client Enhancements

- [Breaking Change] Converted the Web products grid to use the new Editable Grid quick form control which allows inline editing in the Web Client. This change impacts any customizations to the opportunity and sales order grids. The grids were converted from custom forms to quick forms.

- Removed the "Add All Users to Standard Role" function. To add users to the Standard User Role, select the appropriate users in the Users List view and click Add to Role in the User Tasks pane.

- Added additional administrative functionality in the Web Client that allows users with the appropriate security access to create new secured actions to secure elements in the Web Client, manage qualifications for leads, manage resources for activities, and manage standard problems and resolutions.

- Added Check for Duplicates functionality. The Admin can run deduplication on account records to better manage the database. In addition, you can add custom entities to the Check for Duplicates wizard. For instructions, refer to the "Enabling Check for Duplicates for an Entity" topic in the Application Architect help.

- Modified the method of linking help to forms. Refer to the "Linking a Help Topic to a Help Button" topic in the Application Architect for more information.

- Added the ability to access SData feeds directly through the Web Client portal so that it handles authentication. For more information, refer to the "Consuming SData through the Web Client" topic in the Application Architect help.

- Added the Notes/History tab with functionality similar to the Network Client. The Notes/History tab includes a preview panel on the right side of the tab.

- Added a new calendar option named "First day of week". Users can set this option to determine which day starts their week (for example, Sunday or Monday).

- [Breaking Change] New configurations were added to PotentialMatchConfiguration.xml and MergeConfiguration.xml files requiring these files to be replaced during the upgrade. If these files were customized, your customizations will be lost.

Application Architect Enhancements

- Modified the Currency Control to contain three valid enum values for ExchangeRateType (Base Rate, My Rate, and Entity Rate) and two deprecated values (Sales Order Rate and Opportunity Rate). See task 61 for more information.

- [Breaking Change] Created a new assembly named Sage.SalesLogix.Utility.dll and renamed the ImportNameParser class to NameParser (Sage.SalesLogix.Utility.NameParser). The ImportNameParser class has been removed and all references where the ImportNameParser was used have been updated to the new NameParser class. The ImportNameParser class was used in the following locations:
  - Sage.SalesLogix.Services.Import.dll (Sage.SalesLogix.Services.Import.ImportNameParser).
  - Sage.SalesLogix.BusinessRules.dll (Sage.SalesLogix.Import.ImportNameParser).

- Added the DateTime format type to Lookup properties and added a format string property. The format string is currently only used when the PropertyFormat is set to DateTime. The format string uses the same date format strings as the group date formats.

- Added the Decimal Digits property to the Currency and Numeric controls and columns.

- Added the ability to enable field-level security for new entities within the New Entity Wizard.

- Added the Field Level Security Service that automatically resolves security profile issues. When security access is requested for an entity property, and a security profile is not long enough to contain an access entry for that property, all security profiles are reviewed and padded with the necessary number of bytes to hold an access entry for each secured field in the database.

- New quick forms enhancements include the SData Data Source control and the Editable Grid control.

- Improved the process of setting the exchange rate for the Currency control and column. Entity Rate was added to Exchange Rate Type and Sales Order and Opportunity rate types were depreciated.

- Updated some symbols in the expression language documentation to correctly represent inequality as < >.
- Added new endpoints.
- SData enhancements:
  - Set the default on new entities for the SData feed to be enabled. You have the option to prohibit create/update/delete operations in the generated dynamic adapter resource handlers. Individual entity properties can be excluded from the SData payload. Refer to the "Generating SData Feeds for an Entity" topic in the Application Architect Help for more information.
  - Exposed more entities to SData.
- Added the ability to customize the tabs on the What's New page by configuring an XML file. For more information, refer to the "Customizing the What's New Module" topic in the Application Architect help.
- Added the SData processor type to the Mashup Designer. For more information, refer to the "SData Processor Type" topic in the Application Architect help.

Architect Enhancements

- Added the ClickKey read/write property string to the DataGridColumnButton object. You can use it to set a string representing a shortcut, for example "F2", "Alt+E", "Ctrl+Enter".
- Modified the "Automatically add Include Scripts" option to remember its last state when creating a bundle. In previous versions this option was always selected by default.

Changes in Version 7.5.3

In addition to defect fixes, v7.5.3 contains the following new functionality.

Note Upgrades to technology previews are unsupported. If you downloaded and installed a technology preview, the service pack overwrites the functionality and features in that preview may be modified or removed.

Deprecated Components

- The Application Architect Security Explorer has been deprecated and removed in this release. Security for Web users is managed in the Web Client Roles view. See the "What are Roles" topic in the Web Client help for details.

General Enhancements

- Added support for Left joins in some binding scenarios.
- [Breaking Change] Modified various classes and members in the SalesLogix API. Of special note are modifications to methods within the UserManagementService in Sage.SalesLogix.Security and SLXWebUserService in Sage.SalesLogix.Web to return the Sage.Entity.Interfaces.IUser interface object. Previously, user-related methods returned concrete Sage.SalesLogix.Security.User objects. See task 62.

Web Client Enhancements

- Added administrative functionality to the Web Client that allows users with the appropriate security access to manage competitors, departments, lead sources, literature items, pick lists, products, packages, users, teams, and security (roles).
- Added the ability to fulfill, complete, or reject literature requests.

- Added browser agnostic Desktop Integration features for drag-and-drop attachments, drag-and-drop e-mail from Microsoft Outlook, drag-and-drop library files (Administrator only), Outlook Integration using Send SLX and record to history, mail merge, and export to Excel.

  When installing the Desktop Integration module, the old (ActiveX) version of the slmn.dll is unregistered and the new (Desktop Integration) slmn.dll is registered. ActiveX files in a previous version cannot be used concurrently with Desktop Integration features.

  > The Desktop Integration module replaces the ActiveMail CAB.

  If you have a Windows and Web environment, ensure you install the Desktop Integration components AFTER installing the Network Client. When you install the Desktop Integration Module after the Sage SalesLogix Network Client, all SendSLX functionality is handled by the Desktop Manager. If you install the Network Client after downloading the Desktop Integration Module, SendSLX will not work correctly on the Web until you reinstall the Desktop Integration module.

- Added Dashboards which allow you to display and interact with a variety of performance and analysis tools on the Welcome page of the Sage SalesLogix Web Client. You can maintain one or more personalized dashboard tabs and, with the appropriate permissions, you can modify dashboards provided by Sage or customized by your administrator. The following new plugin types were added for dashboards: 35=Dashboard Widgets, 36=Dashboard pages (or tabs).

  > In this release, the legacy Welcome page (Welcome.aspx) is still available in addition to the new dashboard page (home.aspx). Both pages are available in the Application Architect (Project Explorer > Portal Manager > Sage SalesLogix > Pages). If you customized your Welcome page and want to continue to use your legacy version with the new dashboards, you can customize the Navigation Bar to display one or both items. See the "Adding Navigation Groups and Items" topic in the Application Architect help for more information.

Mail Merge Enhancements

- [Breaking Change] Converted the MailMerge.vbs from VBScript to JavaScript.
  - The functionality formerly in the MailMerge.vbs script has been moved into sage-mailmerge-service.js and sage-mailmerge-loader.js.
  - The "Sub MailMergeEngine_OnCustomFieldName" is now handled in sage-mailmerge-service.js in the Sage.MailMergeService.prototype.HandleOnCustomFieldName handler. This handler may have been modified if your customizations include custom mail merge fields.
- [Breaking Change] Deprecated the MailMerge.js script. The functionality in this script has been incorporated into the sage-mailmerge-service.js, sage-mailmerge-context.js, and sage-mailmerge-loader.js scripts.
- [Breaking Change] Converted the SpMailMerge_ClientScript.vbs script from VBScript to JavaScript.
  - The new file name is SpMailMerge.js.
  - Renamed the method called to instantiate a mail merge job from sp_DoMailMergeVB to sp_DoMailMerge.
- Removed the ActiveX object tags. Any script that references the following object tags must be refactored to use the mail merge API:
  - AddressLabels
  - MailMerge
  - MailMergeGUI

- ProgressDlg
- TemplateEditor

> Most object tags were only used by the legacy MailMerge.js script; the MailMergeGUI object tag is the most likely one that may have been used in customizations. Customizations can be refactored from top.MailMergeGUI.%function% to GetMailMergeService().MailMergeGUI().%function%.

- [Breaking Change] The legacy ActiveX Mail Merge API used class named SLXDoc.SLXDocument (SLXDocW in 7.5.2), which in turn exposed the SLXDoc.MailMergeInformation class that was used to create the mail merge request. This has been replaced with the MailMergeInformation class that is exposed from the MailMergeService.

Web Customization Enhancements

The following changes improve the Web customization experience. See the Application Architect Help for details on these changes.

- Added updates to the User Option Service.
- Added secured actions to control access to user interface elements like edit or delete.
- You can create new, edit existing, release, or delete dashboard widgets that display on the Welcome page in the Sage SalesLogix Web Client. Use pre-defined filters for ranges or distinct fields, or create new filters for chart dimensions and metrics.
- Added or modified the following quick form controls and properties:
  - **Hql Data Source control** - allows you to specify an Hql statement to be used as a source of data.
  - **Insert Association action** - gives the ability to check for duplicate associations.
  - **Data Source control** - gives the ability to specify the join direction with the Data is Required property.
  - **ComboBox and RadioGroup controls** - now allow the selected value to be specified through data binding.
  - **Owner control** - use the Multiselect and Types properties to allow multiple selections and to limit the owner types.
  - **TextBox control** - added the TextMode property so that text can be displayed in password mode; that is, characters are replaced by bullets.
  - Added the Applied Security property to several controls and grid columns in support of the Secured actions feature.
  - **PickList control** - behaviors can be overridden in the Pick List Detail view in the Web Client.
- Sage Data (SData) updates.
  - Entity business methods exposed as service operations. For example, /sdata/slx/dynamic/-/accounts/$service/ValidateEmployeeCount.
  - Custom properties included in payloads.
  - Async support for $batch and $service requests, including support for reliable posting.
  - Unlimited depth property chaining in request URLS. For example: /sdata/slx/dynamic/-/contacts('CA2EK0013122')/Account/DivisionalManager/Owner.
  - Dot notation in where and orderby query parameters.
- Updated Help to include:
  - Information on contextual awareness for custom entity Main views. When users navigate from one main view to another, the current group list must be re-built based

on the page the user navigated from. A file to specify contextual awareness was added in version 7.5.2. Information on it was added to the help in 7.5.3.

• An example of adding custom activity group tabs. The Activity List view is a custom form which can be customized to add new group tabs.

Changes in Version 7.5.2

In addition to defect fixes, v7.5.2 contains the following new functionality.

Deprecated Components

• [Breaking Change] The Infragistics2.WebUI.UltraWebNavigator.v7.1.dll component formerly deployed with the Web Client, has been deprecated and removed from this release. If you are using this component in any customizations, you must continue to use your existing version as it will not be updated by Sage SalesLogix.

General Enhancements

• [Breaking Change] Added support for nVelocity and the T4 templating engine. This engine provides intellisense and debugging support. Sage SalesLogix uses the T4 engine for code generation. A set of default entity templates will be maintained in the T4 engine.

If you modified the Sage entity templates (nVelocity), you must recreate your changes in the new T4 templates.

• .Net Framework v3.5 SP1 is required for this release.

• Added support for Microsoft Unity.

• The Sage SalesLogix platform was updated for nHibernate v2.1 and Microsoft Object Builder v2.0.

• New Sage SalesLogix Selection Service.

This service is used to map the selection context to the correct selection request in a Web Client List view. See the API Reference for information on this service.

• You can choose how Send SLX handles attachments. See task 34 for details.

• Navigation to an entity main view in JavaScript has been simplified. A client-side method that accepts any entity type and ID and directs the user to the entity in the appropriate way is now available. See the "Linking to Entity Views with JavaScript" topic in the Application Architect Help for details.

• This release allows you to enable field-selectable Unicode. If you choose to implement Unicode, users can create, view, update, or delete records that are represented using a Unicode character set.

Important Field-selectable Unicode is limited to specific feature areas. Extending the database for Unicode requires a strong knowledge of Sage SalesLogix and its database schema. Before implementing this feature, you should carefully review the "Understanding Unicode" section in the Administrator Help for details. Persons implementing this feature assume full responsibility and understand the risks and limitations associated with Unicode in Sage SalesLogix. Sage strongly recommends you back up your database before making any Unicode changes.

• Intellisync for Sage SalesLogix has increased stabilization and includes SSL Support and portal validation and statistic information in the Personal Web Server.

• You can configure FTP synchronization to use Passive or Active FTP connections. See the "Configuring FTP Sync to Run in Active or Passive Mode" topic in the Administrator Help for details.

Mail Merge Enhancements

- [Breaking Change] Mail merge code has been separated for the Network and Web Clients. This introduces new names for Web DLLs and may impact customizations. See task for details.
- This release installs the new Slxmm.dll on the Web Host. This DLL enables mail merge functionality on all systems even if mail merge was not previously installed.

Web Client Enhancements

See the "What's New in this Release" topic in the Web Client Help for details on these changes.

- Offline Web Client users do not need admin rights to upgrade their machines. See task for details.
- Offline Web Client users can generate reports locally against their remote database. See task 84 for details.
- A new option has been added to groups. Use the Default Lookup Layout option to select the group with the layout you want to use for lookups on each main view.
- Activity Reminders have been updated to allow users to mass-dismiss reminders, mass-delete activities using the Reminders window, and confirm or decline confirmations directly from the Reminders view.
- The behavior of Numeric controls has changed for entries that are less than 1%. In previous versions, if a decimal number was entered in a Numeric control of format type = Percent, Sage SalesLogix multiplied it by 100. For example, if 0.5 was entered, it was assumed to be 50% and stored as 0.5. In this release, entries of less than 1 are recognized as a percent smaller than 1 and are stored accordingly. For example, a value of 0.5 is stored as .005. This change affects new data entry, but does not change data already stored in the database.
- [Breaking Change] Sales Orders in the Web Client have been promoted to a main entity and are now required to be associated to an account instead of an opportunity. To support this change, Web Client users will see the following:
  - The Sales Orders button has been added to the Navigation Bar which opens the Sales Orders List view.
  - The Sales Orders Detail view contains a Sales Orders Snapshot with financial information that can be forwarded in an e-mail message.
  - When a sales order address is modified, a new view opens prompting the user to update matching contact addresses with the change and/or to update any matching open sales orders addresses.

    Sales Orders in the Network Client are unchanged.

Performance Enhancements

The Web Client has improved performance in the following areas:

- [Breaking Change] Combined and condensed JavaScript and CSS files to reduce the number of HTTP requests to the server. You can add your custom JavaScripts and CSS files to the same projects to realize the same benefit. See the "Improving Performance of Custom JavaScript" topic in the Application Architect for details.
- [Breaking Change] The form type from which the Copy control can copy has been limited to summary views. See task 66 for more information.

- [Breaking Change] Refresh behavior has been removed from dialog boxes. In previous releases, the WebDialogService initiated a refresh when a dialog box was closed. To eliminate unnecessary refreshes from dialog boxes that were opened and closed without changes, all refreshes are now handled by the smart part. If you have custom smart parts displayed in dialog boxes that update data displayed in the form behind the dialog, you must change each custom smart part to initiate a refresh after an update. See task 63 for more information.

- Created a cleaner infrastructure for faster page loading, quicker response time, and smaller bandwidth.

- Reduced unnecessary processing by updating panel efficiencies for the dialog workspace, tab workspace, and smart parts.

- Reduced HTML markup page request size and removed inline styling for easier customization and less markup.

- Identified and fixed long running queries.

- When you search for duplicates while adding a new contact, account or lead or converting a lead, incremental indexing before the search is disabled by default. In earlier versions, indexing occurred automatically. In this release, indexing is not performed before the search unless you enable the TurnOnIndexing option in the PotentialMatchConfigurations.xml file. See the "Setting Potential Match Indexing Options" topic in the Application Architect Help for details.

- Import Advanced Options are available to control how indexing is performed before a lead import starts and after each record is inserted. By default, when you run a lead import and search for duplicates, incremental indexing before the import is enabled and updating the index during the import is disabled. In earlier versions, indexing occurred automatically during the import. Indexing will be performed during an import if the user selects the "Check for duplicates contained within the import source file" option in the Lead Import dialog box or the IndexAfterInsert option in the ImportConfigurations.xml file is set to true. See the "Setting Potential Match Indexing Options" topic in the Application Architect Help for details.

- Most cases that required the Use Smart Selection property to be set to False have been resolved. The exception is when a grid shows a calculated field defined as a code snippet property. In this instance, the Use Smart Selection property must be set to False. To locate opportunities to improve performance, review DataSource controls in your customizations and set Use Smart Selection to True except in cases where a code snippet property is shown on the grid. In those cases, consider redefining the property as a calculated field in the Administrator if you want to set Use Smart Selection to True. See the "Use Smart Selection" topic in the Application Architect Help for more information.

- The SLX Indexes for MS SQL.sql script is available if you are running Sage SalesLogix on Microsoft SQL Server. Depending on your database, this script may improve performance by optimizing database indexes.

  > Applying this script is optional. The script is provided "as is" and should be reviewed, along with the Index Changes.pdf document, to determine compatibility with your database.

Web Customization Enhancements

The following changes improve the Web customization experience. See the Application Architect Help for details on these changes.

- [Breaking Change] Portal page inheritance has been modified to derive from the Web portal page. All portal pages should inherit from Sage.Platform.WebPortal.WebPortalPage. Code generation templates have been updated to use this type as the base class to the page. If you have any custom pages that were generated from default portal pages, you must update their inheritance as detailed in task 60. When a portal is compiled, any pages that have not had their base type set correctly may report an error around missing methods such as RegisterSmartPart, RegisterModule. If you see these errors, the portal page base type is incorrect and must be re-mapped.

- [Breaking Change] Classes in the security and address assemblies required modifications. The "Sage SalesLogix API Change List for v7.5.2" document lists the assembly and property changes in this release. Refer to this document to update your custom code.

- A flag has been added to assist troubleshooting data binding errors. This change allows you to see data binding errors that happen at run-time.

- [Breaking Change] Sage SalesLogix is compliant with the 1.0 SData specification. New features include support for batch processing, asynchronous operations, template support, and schema discoverability. The base URL and some payload formatting have changed to comply with this specification. If you have any customizations that consume an SData feed, you must update your URL as detailed in task 67.

- [Breaking Change] The method for registering script references in controls has been changed to follow the Microsoft AJAX Framework standard. If you have a custom control on an AJAX update panel that registers scripts, follow the steps in task 68 to upgrade.

- [Breaking Change] The positioning behavior of dialog boxes in the dialog workspace has been updated for the following:
  - The Top and Left parameters are not required.
  - If the Top and Left parameters are not set, the dialog box will be centered.
  - If the Top and Left parameters are set, but CenterDialog is True, the Top and Left parameters are ignored.
  - If the Top and Left parameters are set to -1, the dialog box will be centered even if CenterDialog is False.

- The Web Client has a customizable logoff/redirect page. When the Web Client times out or a user logs off, you can redirect to another page. See the "Redirecting Web Client Logoff/Timeout" topic in the Application Architect Help for details.

- Added the MySLX API. MySLX is a new library of shortcuts for commonly used objects, methods, and programming tasks. This release introduces the MySLX functionality and provides several initial utility methods. Feedback from our customers and business partners will guide the rapid evolution of this convenient and time-saving developer tool. For more information, refer to the Sage SalesLogix API Reference.

- Security entities are now defined in the model and are extendable. You can add your own properties/columns and business rules and can create instances through the EntityFactory in the same way you can with other entities in the model.

- Asynchronous job support is available for longer running tasks in the Sage SalesLogix Web Client. This support is exposed both as a reusable component (JobService) and a helper class for implementing SData asynchronous operations (SDataAsyncOperation) which uses JobService and adds SData specific implementation.

- You can narrow the scope of lookups on an entity in the Customer Portal by adding a condition in the web.config file. Refer to the Application Architect Help topic "Adding Conditions to Lookups in Customer Portal" for more information.

- The ability to use custom controls in Application Architect quick forms is now available. Refer to the Application Architect Help topic "Creating Custom Controls for Quick Forms" for details.

- Added the ability to set the format string for each control property of type string that is available in data bindings. Refer to the Application Architect Help topic "Bind Control Properties Collection Editor" for details.

- The new Find Orphans utility identifies code snippet orphans that may exist in your code. You can use this utility to find code snippet orphans that you may want to delete. This can help resolve issues you may have building the code snippet library. FindOrphans.exe is installed on the Administrative Workstation in the SalesLogix folder.

- The following DataGrid properties are now available in code snippet actions: SelectedIndex, DataSource, and DataBind() (method).

- New bundle differencing utility (BundleDiff.exe) that creates a list of all items in a bundle and finds differences between two bundle manifests. You can use this utility to preview customizations that may require merging during an upgrade or to see a complete list of items that Sage has updated in this release.

  Sage provides differences bundles that contain Web changes from version 7.5 and later. The bundles are located in the Project Differences folder. You can use these bundles with the BundleDiff utility to view information about Sage changes and how they impact your customizations. See the "BundleDiff.exe" topic in the Application Architect Help for more information.

- A Hide method was added to the tabWorkspace class to allow hiding and showing a tab using code in a module. See the "Hiding/Unhiding a Tab at Run Time" topic in the Application Architect Help for more information.

Network Customization Enhancements

The following changes improve the Sage SalesLogix Client customization experience. See the Architect Help and LAN Developers Reference Help for details on these changes.

- The default lookup criteria (currently "Starts with") has been exposed so you can override it on the LookupEdit control or through the API.

- Added Intellisense support for the ClearContextList and AddToContextList methods.

- Added the new ArrayToString method. This method allows you to convert an array of bytes such as those returned from an ADO blob to a string.

- Added optional properties for the Application.BasicFunctions.LookupItemWithConditionByID and Application.BasicFunctions.ShowMainViewFromLookupWithConditionByID methods.

Changes in Version 7.5.1

In addition to defect fixes, v7.5.1 contains the following new functionality:

- The Web Client activities view has been modified to include a "split" view, filters, and a summary view.

- You can control field level security access to individual fields in Web Client Detail views, List views, groups, and filters. Security profiles are created and maintained in the Administrator for both Network and Web users. See the "Security Profile Manager" and "User Profile Security Tab" topics in the Administrator Help for field level security information and instructions.

- Web Reporting is available for Web Remote Offices.

- Web Reporting has been modified so that one Web Reporting Server supports one Web site. In previous versions one Web Reporting Server could support multiple Web sites.

  Manual configuration of the Web Reporting Server is no longer required. The Report Server Configuration dialog box has been removed from the Application Architect.

- Offline Web Client users can subscribe/unsubscribe to accounts using the What's New view.

- Mail Merge is available for Offline Web Clients. The processing handler for mail merge on the Web has been modified so that HTTP requests are targeted for processing by an ASP.NET assembly instead of an ISAPI module. The Web Client processing remains the same. This update removes the dependency on IIS for mail merge allowing mail merge to run using the Sage SalesLogix Personal Web Server.

  Manual configuration of the Active Mail Server is no longer required. The Active Mail Server Configuration dialog box has been removed from the Application Architect.

  In v7.5.1 and later, Active Mail must be installed on the Web Host. You cannot host Active Mail on a machine other than the Web Host.

- A user survey is embedded in the Web Client that is used to capture data related to user work practice. Web Client users are periodically prompted to participate in the online survey. You can add your own customizable survey for an application version. See the "Configuring the Work Practice Survey" topic in the Application Architect help for more information.

- When creating a bundle manifest in Application Architect, you can set the Web bundle to contain a password that must be entered before the bundle can be installed.

- Lotus Notes v7 and 8 support has been added in Intellisync for SalesLogix.

- The following Application Architect components have been added or updated in this release: Panel control, MultiTab control, Sparkline control for Quick Forms, Mashup Data Source control. The build and deploy process has been improved to allow partial builds or builds of a specific area (such as assemblies). Build speed has also been increased.

- Firefox v3.0x is supported for all Web Client functionality except mail merge, Export to File and Outlook Integration (SendSLX and Record to History).

  Support for mail merge, Export to File, and Outlook Integration on Firefox v3.0x was added in v7.5.3.

Chapter 3
Plugin Changes in this Release

In addition to the new functionality described in Chapter 2, "Changes in this Release", this service pack fixes product defects. This includes defects previously addressed in hot fixes.

Refer to the Fixed Issues List on the Sage SalesLogix Support Portal Web site for all defects fixed in this service pack.

Plugin changes are listed in the following sections by category (new or existing), type (form, script, and so on), and then alphabetically by plugin name within the category and type.

Finding Script Changes

Changes to Sage SalesLogix scripts, and scripts on forms, can be researched using a third-party comparison utility such as Beyond Compare or Microsoft Word. You can use the following example procedure to determine the Sage SalesLogix script changes in this release. Then, use that information to update your custom scripts with the Sage SalesLogix changes, or add your customizations to the Sage SalesLogix script.

To find script changes

1. Apply the upgrade bundle to a test environment.
2. Open the original version of the script or form you want to research in the Architect.
3. Do one of the following:
    *   For a form: Click the **Script** tab, right-click the script, and then click **Select All**.
    *   For a script: Right-click the script, and then click **Select All**.
4. Copy and paste the information to a text editor, such as WordPad.
5. Save the script with the version number in the name.
6. Repeat steps 2 - 5 for the same plugin updated in this release.
7. Open the original plugin version in Microsoft Word (saved in step 5).
8. On the **Tools** menu, click **Compare and Merge Documents**.
9. Browse to and select the updated plugin (saved in step 6) and click **Merge**.
10. View the code changes and determine how to merge the Sage SalesLogix changes with your customizations.

New Tables

Table Name	Purpose	7.5.1	7.5.2	7.5.3	7.5.4	8.0
ACCOUNTOPERATINGCOMP	Stores operating company information from the accounting company. This table has a many-to-many relationship between the account and operating companies. Added for accounting integration functionality.				✓	
APPIDMAPPING	Stores the GUID and EndPoint URL for Integration Contract synchronization. Added for accounting integration functionality.				✓	
COMMODITYGROUP	Stores commodities and products grouped by category. Added for accounting integration functionality.				✓	

Table Name	Purpose	7.5.1	7.5.2	7.5.3	7.5.4	8.0
COUNTRYCODEMAPPING	Maps the Address Country to an ISO Country Code. Added for accounting integration functionality.				✓	
DEDUPJOB	Stores details about the running de-duplication job.				✓	
DEDUPRESULT	Stores the results as Sage SalesLogix processes a de-duplication job.				✓	
ERPEMAILADDRESS	Stores the e-mail addresses for any entity (Account, Contact) and maps the e-mail columns in these tables to rows in itself. Added for accounting integration functionality.				✓	
ERPPHONENUMBER	Stores the phone numbers for any entity (Account, Contact) and maps the phone number columns in these tables to rows in itself. Added for accounting integration functionality.				✓	
ERPSALESORDER	An extension table for ERP specific fields. Added for accounting integration functionality.				✓	
ERPTRADINGACCOUNT	Stores the additional trading account (company) information from the accounting system. Added for accounting integration functionality.				✓	
QRTZ_BLOB_TRIGGERS						✓
QRTZ_CALENDARS						✓
QRTZ_CRON_TRIGGERS						✓
QRTZ_FIRED_TRIGGERS						✓
QRTZ_JOB_DETAILS						✓
QRTZ_LOCKS						✓
QRTZ_PAUSED_TRIGGER_GRPS						✓
QRTZ_SCHEDULER_STATE						✓
QRTZ_SIMPLE_TRIGGERS						✓
QRTZ_SIMPROP_TRIGGERS						✓
QRTZ_TRIGGERS						✓
REMOTEAUTOCLEANUP						✓
SECUREDACTION	Stores all secured actions.			✓		
SECUREDACTIONROLE	Stores the roles associated to a secured action.			✓		
SLXCONTRACTSYNCLOG	Stores the values for accounting integration that indicate which records have been inserted or deleted since the last synchronization.				✓	
SLXLOCATION	Stores the warehouse or location that a commodity can be shipped from. Added for accounting integration functionality.				✓	
SLXLOGITEM	Stores application log entries from various integration sources (such as synchronization). Added for accounting integration functionality.				✓	
SLXPRICE	Stores commodities and their prices. Added for accounting integration functionality.				✓	
SLXPRICELIST	Stores the grouping of commodities and their prices. Added for accounting integration functionality.				✓	

Table Name	Purpose	7.5.1	7.5.2	7.5.3	7.5.4	8.0
SYNCDIGEST	Stores which tables Sage SalesLogix needs to send to a target. Added for accounting integration functionality.				✓	
SYNCJOB	Stores information specific to the instance of an executed synchronization cycle. Added for accounting integration functionality.				✓	
SYNCRESULT	Stores the results from a target endpoint when the endpoint attempts to apply changes. Added for accounting integration functionality.				✓	
UNITOFMEASURE	Stores unit of measure information for products. Added for accounting integration functionality.				✓	

New Fields

Table Name	Field Name	Purpose	7.5.1	7.5.2	7.5.3	7.5.4	8.0
ACCOUNT	APPID	Maps the endpoint of the system that made the last change. Added for accounting integration functionality.				✓	
ACCOUNT	CREATESOURCE	Stores a string value of the system that created the account. Added for accounting integration functionality.				✓	
ACCOUNT	GLOBALSYNCID	Stores the global identifier for all applications (for example, Sage SalesLogix and an accounting system). This value must be present in both systems for sync to match up the records and link them between systems. Added for accounting integration functionality.				✓	
ACCOUNT	LASTERPSYNCUPDATE	Stores the date and time the account was last updated by accounting integration synchronization. Added for accounting integration functionality.				✓	
ACCOUNT	PRIMARYOPERATINGCOMPID	Stores the GUID that maps to an AppID in the APPIDMAPPING table. Added for accounting integration functionality.				✓	
ACCOUNT	PROMOTEDTOACCOUNTING	Stores a True or False value to indicate if the account has been promoted to the accounting system. Added for accounting integration functionality.				✓	
ACCOUNT	TICK	Stores an integer value that is incremented by Sage SalesLogix when a change is made to a record. Added for accounting integration functionality.				✓	
ACCOUNTPRODUCT	CREATESOURCE	Identifies the application that created the record.					✓
ACTIVITY	ATTACHMENTCOUNT	Stores the number of attachments for the activity item.		✓			

Table Name	Field Name	Purpose	7.5.1	7.5.2	7.5.3	7.5.4	8.0
ACTIVITY	CREATESOURCE	Identifies the application that created the record.					✓
ACTIVITY	LOCATION	Stores the location assigned for an activity item.					✓
ADDRESS	ADDRESSTYPE						✓
ADDRESS	APPID	Maps the endpoint of the system that made the last change. Added for accounting integration functionality.				✓	
ADDRESS	CREATESOURCE	Identifies the application that created the record.					✓
ADDRESS	ERPNAME	Stores a name for the portal address (for example, Billing Address). Added for accounting integration functionality.				✓	
ADDRESS	GLOBALSYNCID	Stores the global unique identifier for accounting integration.				✓	
ADDRESS	PRIMARYADDRESS						✓
ADDRESS	TICK	Stores an integer value that is incremented by Sage SalesLogix when a change is made to a record. Added for accounting integration functionality.				✓	
APPIDMAPPING	ENABLEEMAILAUTH						✓
APPIDMAPPING	ISSYSTEMENDPOINT						✓
ATTACHMENT	CREATESOURCE	Identifies the application that created the record.					✓
ATTACHMENT	SALESORDERID	Assigning a SalesOrderId associates this attachment to the SalesOrder.		✓			
CALCULATEDFIELDDATA	SORTORDER	Added for performance improvements. For Sage use only.		✓			
CAMPAIGN	CREATESOURCE	Identifies the application that created the record.					✓
COMPETITOP	CREATESOURCE	Identifies the application that created the record.					✓
CONTACT	ACTIVEFLAG	Stores a True or False value to indicate if the contact is an active contact. Added for accounting integration functionality.				✓	
CONTACT	APPID	Maps the endpoint of which system made the last change. Added for accounting integration functionality.				✓	
CONTACT	COMPANYCONTEXT	Stores the description of the contact's reference with a company (if any). For example, employee or subcontractor. Added for accounting integration functionality.				✓	
CONTACT	CREATESOURCE	Identifies the application that created the record.					✓

Table Name	Field Name	Purpose	7.5.1	7.5.2	7.5.3	7.5.4	8.0
CONTACT	GENDER	Stores the contact's gender (Male, Female, or Not Specified). Added for accounting integration functionality.				✓	
CONTACT	GLOBALSYNCID	Stores the global unique identifier for accounting integration.				✓	
CONTACT	MARTIALSTATUS	Stores the contact's martial status.Added for accounting integration functionality.				✓	
CONTACT	NATIONALITY	Stores the contact's nationality. Added for accounting integration functionality.				✓	
CONTACT	PREFERREDNAME	Stores the contact's preferred name. Added for accounting integration functionality.				✓	
CONTACT	PRIMARYLANGUAGE	Stores the contacts spoken language. Added for accounting integration functionality.				✓	
CONTACT	TICK	Stores an integer value that is incremented by Sage SalesLogix when a change is made to a record. Added for accounting integration functionality.				✓	
DEFECT	CREATESOURCE	Identifies the application that created the record.					✓
ERPEMAILADDRESS	CREATESOURCE	Identifies the application that created the record.					✓
ERPPHONENUMBER	CREATESOURCE	Identifies the application that created the record.					✓
EVENT	CREATESOURCE	Identifies the application that created the record.					✓
EVENT	LOCATION	Stores the location assigned for an event item.					✓
HISTORY	ATTACHMENTCOUNT	Stores the number of attachments for the history item.		✓			
HISTORY	CREATESOURCE	Identifies the application that created the record.					✓
HISTORY	LOCATION	Stores the location assigned for a history item.					✓
INDEXDEFINITION	ISSECURE						✓
LEAD	CREATESOURCE	Identifies the application that created the record.					✓
LEAD_ADDRESS	CREATESOURCE	Identifies the application that created the record.					✓
LEADSOURCE	CREATESOURCE	Identifies the application that created the record.					✓
LIBRARYDOCS	CREATESOURCE	Identifies the application that created the record.					✓
OPPORTUNITY	CREATESOURCE	Identifies the application that created the record.					✓

Table Name	Field Name	Purpose	7.5.1	7.5.2	7.5.3	7.5.4	8.0
OPPORTUNITY	OVERRIDESALESPOTENTIAL	Stores a True or False value to indicate if auto calculation is not used to calculate sales potential. Added for accounting integration functionality.				✓	
OPPORTUNITY	OPERATINGCOMPID	Stores the ID of the operating company that created the opportunity. Added for accounting integration functionality.				✓	
OPPORTUNITY	PRICELISTID	Stores the foreign key relationship to the SLXPRICELIST table. Added for accounting integration functionality.				✓	
OPPORTUNITY_CONTACT	CREATESOURCE	Identifies the application that created the record.					✓
OPPORTUNITY_PRODUCT	CREATESOURCE	Identifies the application that created the record.					✓
PACKAGEPRODUCT	PARENTPRODUCTID	Stores the hierarchy relationship within products to support kits from the accounting system. Added for accounting integration functionality.				✓	
PACKAGES	DESCRIPTION	Stores the package description.			✓		
PACKAGES	STATUS	Stores the package status.			✓		
PRODUCT	ACTIVEFLAG	Stores a True or False value to indicate if the commodity is an active commodity. Added for accounting integration functionality.				✓	
PRODUCT	APPID	Maps the endpoint of which system made the last change. Added for accounting integration functionality.				✓	
PRODUCT	CLASSIFICATION	Stores the classification of the commodity. Added for accounting integration functionality.				✓	
PRODUCT	COMMODITYGROUPID	Stores the ID of the commodity group. Added for accounting integration functionality.				✓	
PRODUCT	COMMODITYTYPE	Stores the type enumeration value for the product as defined in the GCRM Contract. Added for accounting integration functionality.				✓	
PRODUCT	CREATESOURCE	Identifies the application that created the record.					✓
PRODUCT	GLOBALSYNCID	Stores the global unique identifier for accounting integration.				✓	
PRODUCT	SELLINGALLOWEDFLAG	Stores a Yes or No value to indicate of the commodity can be used on sales orders and quotations. Added for accounting integration functionality.				✓	
PRODUCT	SELLINGUOMID	Stores the foreign key to the UNITOFMEASURE table. Added for accounting integration functionality.				✓	

Table Name	Field Name	Purpose	7.5.1	7.5.2	7.5.3	7.5.4	8.0
PRODUCT	SELLINGUOMNUMBER	Stores the number of items in a commodity group. Added for accounting integration functionality.				✓	
PRODUCT	TICK	Stores an integer value that is incremented by Sage SalesLogix when a change is made to a record. Added for accounting integration functionality.				✓	
PRODUCT	UNITOFMEASUREID	Stores the foreign key to the UNITOFMEASURE table. Added for accounting integration functionality.				✓	
RESOURCELIST	DESCRIPTION						✓
RESOURCELIST	ISLOCATION						✓
RESYNCTABLEDEFS	DELAYDAYS						✓
RESYNCTABLEDEFS	RESTRICTDATA						✓
RMA	CREATESOURCE	Identifies the application that created the record.					✓
RMAADDRESS	CREATESOURCE	Identifies the application that created the record.					✓
RMAADDRESS	SALUTATION						✓
SALESORDER	ACCOUNTMANAGERID	Stores the assigned seccodeid.		✓			
SALESORDER	ACTIVEFLAG	Stores a True or False value to indicate if the sales order is active. Added for accounting integration functionality.				✓	
SALESORDER	APPID	Maps the endpoint of which system made the last change. Added for accounting integration functionality.				✓	
SALESORDER	CREATESOURCE	Identifies the application that created the record.					✓
SALESORDER	DATEPROMISED	Stores the date the sales order was promised for completion.		✓			
SALESORDER	DISCOUNT	Stores a decimal value which allows a discount to be applied to the sales order. The discount is calculated based on the OrderTotal property of the sales order (this should comprise the total of all the SALEORDERITEMS (products). This is different than the discount in the SALESORDERITEMS table which allows a discount to be applied to an individual line item. If desired, both discounts can apply.		✓			
SALESORDER	DISCOUNTTOTAL	Stores the total discount amount used on the sales order. Added for accounting integration functionality.				✓	
SALESORDER	DUEDATE	Stores the date the sales order is due to the customer. Added for accounting integration functionality.				✓	

Table Name	Field Name	Purpose	7.5.1	7.5.2	7.5.3	7.5.4	8.0
SALESORDER	EXCHANGERATE	Stores the exchange rate for the sales order.		✓			
SALESORDER	EXCHANGERATEDATE	Stores the last date the exchange rate was changed.		✓			
SALESORDER	EXCHANGERATELOCKED	Allows for mass updates to occur for exchange rate changes. When the value is True, the sales order is ignored during an update. Since the Administrator does not support mass updates for sales orders this field is not exposed in the UI. The default setting is True.		✓			
SALESORDER	GLOBALSYNCID	Stores the global unique identifier for accounting integration.				✓	
SALESORDER	GRANDTOTAL	Stores the grand total of the sales order. This field is used for groups and reporting.		✓			
SALESORDER	ISQUOTE	Stores a boolean value indicating if the sales order is a quote. Added for accounting integration functionality.				✓	
SALESORDER	OPERATINGCOMPID	Stores the ID of the operating company that created the sales order. Added for accounting integration functionality.				✓	
SALESORDER	PRICELISTID	Stores the foreign key relationship to the SLXPRICELIST table. Added for accounting integration functionality.				✓	
SALESORDER	REQUESTEDBY	Stores the contact ID that requested the sales order.		✓			
SALESORDER	SECCODEID	Stores the "owner" (Account seccodeid) of the sales order.		✓			
SALESORDER	TICK	Stores an integer value that is incremented by Sage SalesLogix when a change is made to a record. Added for accounting integration functionality.				✓	
SALESORDER	TAXTOTAL	Stores the total tax amount used on the sales order. Added for accounting integration functionality.				✓	
SALESORDERADDRESS	APPID	Maps the endpoint of which system made the last change. Added for accounting integration functionality.				✓	
SALESORDERADDRESS	CREATESOURCE	Identifies the application that created the record.					✓
SALESORDERADDRESS	GLOBALSYNCID	Stores the global unique identifier for accounting integration.				✓	
SALESORDERADDRESS	SALUTATION						✓
SALESORDERADDRESS	TICK	Stores an integer value that is incremented by Sage SalesLogix when a change is made to a record. Added for accounting integration functionality.				✓	

Table Name	Field Name	Purpose	7.5.1	7.5.2	7.5.3	7.5.4	8.0
SALESORDERITEMS	APPID	Maps the endpoint of which system made the last change. Added for accounting integration functionality.				✓	
SALESORDERITEMS	CREATESOURCE	Identifies the application that created the record.					✓
SALESORDERITEMS	CALCULATEDPRICE	Stores the items price * discount. The main purpose is for binding.		✓			
SALESORDERITEMS	COMMODITYTYPE	Stores the type enumeration value for the product as defined in the GCRM Contract. Added for accounting integration functionality.				✓	
SALESORDERITEMS	GLOBALSYNCID	Stores the global unique identifier for accounting integration.				✓	
SALESORDERITEMS	ITEMLOCKED	Stores a True or False value to indicate if the sales order item's price is locked. Added for accounting integration functionality.				✓	
SALESORDERITEMS	LINENUMBER	Stores the sales order line number used for sequencing and ordering of lines. Added for accounting integration functionality.				✓	
SALESORDERITEMS	LINETYPE	Stores the enumerator indicating the type of sales order line. Added for accounting integration functionality.				✓	
SALESORDERITEMS	PRICEDETAILDESCRIPTION	Stores a price description returned by the pricing service. Added for accounting integration functionality.				✓	
SALESORDERITEMS	PRICEDETAILNOTES	Stores price break information from the pricing service. For example, a price break may occur if purchasing additional quantities results in a lower price. Added for accounting integration functionality.				✓	
SALESORDERITEMS	SLXLOCATIONID	Stores a reference to the warehouse the item will be shipped from. Added for accounting integration functionality.				✓	
SALESORDERITEMS	TICK	Stores an integer value that is incremented by Sage SalesLogix when a change is made to a record. Added for accounting integration functionality.				✓	
SALESORDERITEMS	UNITOFMEASUREID	Stores the foreign key to the UNITOFMEASURE table. Added for accounting integration functionality.				✓	
SYSTEMINFO	APPID	Stores a unique ID for each Sage SalesLogix database so that other applications can synchronize using Integration Contract. Added for accounting integration functionality.				✓	

Table Name	Field Name	Purpose	7.5.1	7.5.2	7.5.3	7.5.4	8.0
SYSTEMINFO	LOCKOUTTIMEOUT	Stores a integer value for how long a user is locked out after entering incorrect password while attempting to log into Sage SalesLogix.					✓
SYSTEMINFO	LOGINATTEMPTTHRESHOLD	Number of times a user can attempt to log into Sage SalesLogix before being locked out.					✓
SYSTEMINFO	SENDSLXSAVEMSGFILES	Stores the "T" or "F" value indicating if the option to attach MSG files for e-mail activity is enabled.		✓			
SYSTEMINFO	UNICODE	Stores the "T" or "F" value indicating if Unicode is enabled.		✓			
TICKET	CREATESOURCE	Identifies the application that created the record.					✓
TICKETACTIVITY	CREATESOURCE	Identifies the application that created the record.					✓
UNITOFMEASURE	CREATESOURCE	Identifies the application that created the record.					✓
USER_ACTIVITY	CREATESOURCE	Identifies the application that created the record.					✓
USERNOTIFICATION	CREATESOURCE	Identifies the application that created the record.					✓
USERSECURITY	FAILEDATTEMPTDATE	Stores Date/Time of last failed attempt for user to log into Sage SalesLogix.					✓
USERSECURITY	LOGINATTEMPTS	Stores number of failed attempts for user to log into Sage SalesLogix. Resets to '0' once user logs into Sage SalesLogix.					✓
VIRTUALFILESYSTEM	ISCOMPRESSED	Indicates whether the data is compressed (zip) in the record.	✓				

New Forms

Form Name	7.5.1	7.5.2	7.5.3	7.5.4	8.0
System: Add Edit Sales Order Address • This view is called from the Add/Edit Sales Order view. To support sales orders as a main entity, this view binds to a sales order address. The previous form was bound to a standard address.		✓			
System: Update Addresses		✓			

New Groups

Group Name	7.5.1	7.5.2	7.5.3	7.5.4	8.0
ACO ACCOUNT: ImportedAccounts • Added for accounting integration functionality.				✓	

Group Name	7.5.1	7.5.2	7.5.3	7.5.4	8.0
ACO ACCOUNT: PromotedAccounts • Added for accounting integration functionality.				✓	
ACO OPPORTUNITY: All Open			✓		
ACO OPPORTUNITY: My Closed Opportunities			✓		
ACO Opportunity: My Open Opportunities			✓		
ACO OPPORTUNITY: My Pipeline			✓		
ACO OPPORTUNITY: My Top Opportunities			✓		
AREACATEGORYISSUE: All					✓
COMPETITOR: All Competitors			✓		
DEDUPJOB: AllDeDupJobs				✓	
DEDUPRESULT: AllDeDupResults				✓	
DEFECTACTIVITYRATE: AllRates					✓
DEPARTMENTVIEW: All Departments			✓		
HISTORY: All History			✓		
HISTORY: Completed Activities			✓		
HISTORY: My Completed Activities			✓		
HISTORY: My Notes			✓		
LEADSOURCE: AllLeadSources			✓		
LITERATURE: All Literature Items			✓		
LITREQUEST: All Open			✓		
PACKAGES: All Packages			✓		
PICKLISTVIEW: All Pick Lists			✓		
PRODUCT: All Products			✓		
PRODUCT: Available Products			✓		
PRODUCT: Discontinued Products			✓		
QUALIFICATION_CATEGORY: All Qualifications				✓	
RESOURCELIST: All Resources				✓	
ROLE: All Roles			✓		
SALESORDER: All Sales Orders		✓			
SALESORDER: My Sales Orders		✓			
SALESORDER: Quotes		✓			
SECUREDACTION: AllSecuredActions				✓	
SYNCJOB: AllSyncJobs				✓	
SYNCJOB: Latest Sync Jobs				✓	
SYNCRESULTS: AllSyncResults				✓	
TEAMVIEW: All Teams			✓		

Group Name	7.5.1	7.5.2	7.5.3	7.5.4	8.0
TICKETACTIVITYRATE: AllTicketActivityRates					✓
TICKETAREAOWNER: All					✓
USERSECURITY: AllUsers			✓		
USERSECURITY: ConcurrentUsers			✓		
USERSECURITY: NetworkUsers			✓		
USERSECURITY: RemoteUsers			✓		
USERSECURITY: RetiredUsers			✓		
USERSECURITY: Templates			✓		
USERSECURITY: Users With Failed Logins					✓
USERSECURITY: WebUsers			✓		
USERSECURITY: WebViewerUsers			✓		

New Joins

Join Name	7.5.1	7.5.2	7.5.3	7.5.4	8.0
COMPETITOR/ACCOUNTID to ACCOUNT/ACCOUNTID			✓		

New Pick Lists

Pick List Name	7.5.1	7.5.2	7.5.3	7.5.4	8.0
Address Type					✓
IC Account Type				✓	
IC Company Context				✓	
IC Gender				✓	
IC Sales Order Status				✓	
Report Families				✓	

New Reports

Report Name	7.5.1	7.5.2	7.5.3	7.5.4	8.0
Reports Sales Order: Sales Order Detail				✓	

New Widgets

Widgets are plugins you can add to a Web dashboard.

Widget Name	7.5.1	7.5.2	7.5.3	7.5.4	8.0
Dashboard Page System: My Dashboard			✓		
Dashboard Page System: Sales			✓		
Dashboard Widget System: Bar Chart			✓		
Dashboard Widget System: Column Chart			✓		
Dashboard Widget System: Default			✓		
Dashboard Widget System: Funnel Chart			✓		
Dashboard Widget System: Group List			✓		
Dashboard Widget System: Line Chart			✓		
Dashboard Widget System: Links			✓		
Dashboard Widget System: Pie Chart			✓		
Dashboard Widget System: Recently Viewed			✓		
Dashboard Widget System: Todays Activities			✓		
Dashboard Widget System: Welcome			✓		

Changes to Existing Main Views

Main View Name / Change	7.5.1	7.5.2	7.5.3	7.5.4	8.0
System: Opportunity Details • For the View: TMainAXForm control TPopupMenu property, set the &ViewList Action to Function and Argument to View: ListDetail.				✓	
System: Ticket Details • For the GroupsPane: TGroupsPane control, set the OnPopupMenuPopup event to GroupsPanePopupMenuPopup. Modified the script on the form.		✓			

Changes to Existing Forms

Form Name / Change	7.5.1	7.5.2	7.5.3	7.5.4	8.0
Account: Associations • (7.5.4)Modified the Width of controls on the form. Modified the script on the form.				✓	
Account: Attachments • (7.5.1) For the grdAttach: TDataGrid control, modified the SQL property.	✓				
Account: Contacts • (7.5.2) For the frmContacts: TAXForm, set Width to 257. For the grdContacts: TDataGrid control, set Width to 245. Modified the script on the form. • (7.5.4) Modified the Height and Width of controls on the form. Modified the script on the form.		✓		✓	
Account: Details • (7.5.4) Modified the script on the form.				✓	

Form Name / Change	7.5.1	7.5.2	7.5.3	7.5.4	8.0
Account: Notes • (7.5.4) Modified the Width of controls on the form. Modified the script on the form.				✓	
Account: Opportunities • (7.5.4) Modified the Height and Width of controls on the form. Modified the script on the form.				✓	
Campaign: Stages Tasks • (7.5.2) For the grdStage Task: TDataGrid control, modified the Height and Width. Modified the script on the form. • (7.5.4) For the grdStageTask: TDataGrid control, changed the NeededDate column type to a Date column. Modified the script on the form. • (8.0) Modified the script on the form.		✓		✓	✓
Campaign: Targets • (7.5.1) Modified the Height and Width of controls on the form. • (7.5.2) For the grdTargets: TDataGrid control, set DefaultRowHeight to 17, Sortable to False, and the OnFormatColumnText event to grdTargetsFormatColumn. Modified the Width of controls on the form. Modified the script on the form. • (7.5.4) For the grdTargets: TDataGrid, set Sortable to True. • (8.0) Removed all Impulse pieces.	✓	✓		✓	✓
Contact: Attachments • (7.5.1) For the grdAttach: TDataGrid control, modified the SQL property.	✓				
Contact: Details • (8.0) Added the lblPreferredName: TLabel, lblGender: TLabel, lblMatiatalStatus: TLabel, lblCompanyContext: TLabel, lblNationality: TLabel, lblLanguage: TLabel, TtxtPreferredname: TEdit, pklGender: TPickList, txtMaritalStatus: TEdit, pklCompanyContext: TPickList, txtNationality: TEdit, and txtLanguage: TEdit controls.					✓
Contact: Notes • (7.5.4) Modified the Width of controls on the form. Modified the script on the form.				✓	
Contact: Tickets • (7.5.2) Modified the Width of controls on the form. Modified the script on the form.		✓			
Contract: Attachments • (7.5.1) For the grdAttach: TDataGrid control, modified the SQL property.	✓				
Contract: Covered Assets • (7.5.1) Modified the script on the form.	✓				
Contract: Tickets • (7.5.1) Modified the script on the form.	✓				
Defect: Attachments • (7.5.1) For the grdAttach: TDataGrid control, modified the SQL property.	✓				
Defect: Tickets • (7.5.2) Modified the Width of controls on the form. For the txtSubject: TEdit control, set Text to blank.		✓			
Lead: Attachments • (7.5.1) For the grdAttach: TDataGrid control, modified the SQL property.	✓				
Lead: Notes • (7.5.4) Modified the script on the form.				✓	
Opportunity: Attachments • (7.5.1) For the grdAttach: TDataGrid control, modified the SQL property.	✓				

Form Name / Change	7.5.1	7.5.2	7.5.3	7.5.4	8.0
Opportunity:Contacts • (7.5.1) Modified the script on the form.	✓				
Opportunity:Order Details • (7.5.2) For frmOrderDetails:TAXForm, set Caption to Sales Orders. For the grdSalesOrders:TDataGrid, set Height to 379, and Width to 285.		✓			
Opportunity:Products • (8.0) Modified the script on the form.					✓
Opportunity:Sales Process • (7.5.1) Modified the script on the form.	✓				
RMA:Attachments • (7.5.1) For the grdAttach:TDataGrid control, modified the SQL property.	✓				
RMA:Return Details • (7.5.1) Modified the script on the form.	✓				
Sales Dashboard:Open Opportunities Analysis View • (7.5.2) Modified the Width of controls on the form. For the grdData:TDataGrid and grdFilters:TDataGrid controls, set DefaultRowHeight to 18. Modified the script on the form.		✓			
Sales Dashboard:Opportunity Detail Analysis View • (7.5.2) Modified the Height and Width of controls on the form. For the grdFilters:TDataGrid control, set DefaultRowHeight to 18. Modified the script on the form.		✓			
Sales Dashboard:Pipeline Status Analysis View • (7.5.2) Modified the Height and Width of controls on the form. For the grdData:TDataGrid control, set DefaultRowHeight to 18, and Width to 192. For the grdFilters:TDataGrid control, set DefaultRowHeight to 18. Modified the script on the form.		✓			
Sales Dashboard:Product Sales Analysis View • (7.5.4) Modified the Width of controls on the form. Modified the script on the form.				✓	
Sales Dashboard:Product Sales Potential Analysis View • (7.5.2) Modified the Width of controls on the form. Modified the script on the form.		✓			
Sales Dashboard:Sales Analysis View • (7.5.4) Modified the Width of controls on the form. Modified the script on the form.				✓	
Sales Dashboard:Sales Dashboard Detail • (8.0) Modified the Color of edtMyCalendar and edtMyTasks controls.					✓
Sales Dashboard:Win Rate Analysis View • (7.5.4) Modified the Width of controls on the form. Modified the script on the form.				✓	
System:Activity Details View • (7.5.1) For the lveAccount:TLookupEdit control, set LookupRestrictAlways to blank, and LookupRestrictField to ACCOUNTID. For the lveOpportunity:TLookupEdit and lveTicket:TLookupEdit controls, set LookupRestrictAlways to blank, LookupRestrictField to ACCOUNTID, and LookupRestrictOp to =. Modified the script on the form. • (7.5.2) For the lveLeader:TLookupEdit control, set ReadOnlyEditor to True. Modified the script on the form. • (8.0) Added the lblLocation:TLabel, edtLocation:TEdit controls.	✓	✓			✓
System:Activity Preview • (8.0) Added lblLocationLabel:TLabel and lblLocation:TLabel controls.					✓
System:Add Campaign Products • (8.0)					✓

Form Name / Change	7.5.1	7.5.2	7.5.3	7.5.4	8.0
System: Add Edit Address • (7.5.2) Modified the script on the form.		✓			
System: Add Edit Campaign Stage • (7.5.1) For the txtDescription: TEdit control, set MaxLength to 64. • (8.0) Modified the color of the Panel2 control.	✓				✓
System: Add Edit Product • (8.0)					✓
System: Add Edit Response • (7.5.1) Modified the script on the form. • (8.0) Removed the lblClicks and tbGrids controls.	✓				✓
System: Add Edit Sales Order • (7.5.1) Modified the script on the form. • (7.5.2) Added the dteExchangeRateDate: TDateTimeEdit, lblAcctManager: TLabel, lveAcctManager: TLookupEdit, txtExchangeRate: TEdit, txtExchangeRateLocked: TEdit, txtGrandTotal: TEdit, txtOrderTotal: TEdit, and txtSecCodeID: TEdit controls. Removed the txtBillSOAddressID: TEdit and txtShipSOAddressID: TEdit controls. Modified the TabOrder and location of controls on the form. For frmSalesOrder: TAXForm, set HelpContext to 75215500, OnAfterPost to AXFormAfterPost, and OnCloseQuery to blank. For the grpBilling: TGroupBox and grpShipping: TGroupBox controls, set Top to 103. For the grpComments: TGroupBox control, set Align to alBottom, Top to 507, and Width to 577. For the grpProducts: TGroupBox control, set Align to alBottom, Top to 279, and Width to 577. For the grpTop: TGroupBox control, set Align to alTop, Height to 94, Left to 0, Top to 0, and Width to 577. For the pklOrderType: TPickList control, set ShowHint to True, and Top to 38. For the pklStatus: TPickList control, set TabStop to True. Modified the script on the form. • (7.5.3) Modified the Width of controls on the form. Modified the script on the form.	✓	✓	✓		
System: Add Edit Stage Task • (8.0) Modified the color of the pnlBudget and pnlAssignment controls.					✓
System: Add Edit Ticket Activity • (7.5.1) Modified the Width of controls on the form. Modified the script on the form. • (7.5.2) Modified the script on the form.	✓	✓			
System: Add New Contact Account • (7.5.2) Modified the script on the form. • (8.0) Modified the color of the lblContactInfo and lblAccountInfo controls.		✓			✓
System: Add Opportunity Product • (8.0) Modified the script on the form.					✓
System: Add Targets From Group • (7.5.1) Modified the script on the form.	✓				
System: Campaign Detail • (7.5.2) Modified the Width of Label controls on the form. Modified the script on the form.		✓			
System: Close Opportunity • (7.5.1) Modified the Width of controls on the form. Modified the script on the form.	✓				
System: Contact Detail • (7.5.4) Modified the script on the form.				✓	
System: Contract Detail • (7.5.4) Modified the script on the form.				✓	

Form Name / Change	7.5.1	7.5.2	7.5.3	7.5.4	8.0
System: Convert Lead • (8.0) Modified the script on the form.					✓
System: Create Group From Targets • (7.5.2) For frmCreateCampaignGroup:TAXForm, set the OnShow event to AXFormShow. Modified the Width of controls on the form. Modified the script on the form.		✓			
System: Defect Detail • (7.5.2) Modified the script on the form.		✓			
System: Email - Send Ticket Information • (7.5.1) Modified the script on the form. • (7.5.4) Modified the script on the form.	✓			✓	
System: History Details View • (7.5.1) For the cmdMeeting:TButton control, set Tag to 100. For the cmdPhone:TButton control, set Tag to 200. For the cmdToDo:TButton control, set Tag to 300. For the lveContact:TLookupEdit control, set LookupRestrictAlways to blank, LookupRestrictField to ACCOUNTID, LookupRestrictOp to =, ReadOnlyEditor to True, and TabStop to True. Modified the script on the form. • (7.5.2) For the lveAccount:TLookupEdit, lveOpportunity:TLookupEdit, and lveTicket:TLookupEdit controls, set ReadOnlyEditor to True. • (7.5.4) Modified the script on the form. • (8.0) Added lblLocation:TLabel and edtLocation:TEdit. Modified the script on the form.	✓	✓		✓	✓
System: Import Leads • (8.0) Modified the script on the form.					✓
System: Import Leads Options • (8.0) Modified the script on the form.					✓
System: Insert Campaign • (7.5.1) Modified the Width of controls on the form. Modified the script on the form. • (8.0) Modified the Color of the pnlTargets and pnlStages controls.	✓				✓
System: Insert New Ticket • (7.5.1) Added the edtDescription:TEdit and edtResolution:TEdit controls. For the memComments:TMemo control, set Height to 169 and Width to 79. For the memDescription:TMemo control, set Width to 151 and OnExitControl to memDescriptionExitControl. For the memResolution:TMemo control, set Width to 151 and OnExitControl to memResolutionExitControl. Modified the script on the form. • (8.0) modified the Color of the lblDetails and lblComments controls.	✓				✓
System: Insert Opportunity • (7.5.1) Modified the script on the form. • (7.5.4) Modified the script on the form. • (8.0) Modified the Color of the lblProduct, lblCompetitors, and lblContacts controls. Modified the script on the form.	✓			✓	✓
System: Lead Detail • (7.5.1) Modified the script on the form. • (8.0) Modified the script on the form.	✓				✓
System: Manage Alternate Address • (7.5.1) Modified the script on the form.	✓				
System: Manage Customer Service Defaults • (8.0) Modified the script on the form.					✓

Form Name / Change	7.5.1	7.5.2	7.5.3	7.5.4	8.0
System:Manage Targets • (7.5.2) For the grdTargets:TDataGrid control, set DefaultRowHeight to 18 and Sortable to False. For the lblCreateDateTo:TLabel control, set Width to 9. For the lblInclude:TLabel control, set Width to 35. For the lblSelect:TLabel control, set Width to 309. Modified the script on the form. • (7.5.3) Modified the script on the form. • (7.5.4) For the grdTargets:TDataGrid, set Sortable to False. Added the lblProgress:TLabel control. Modified the script on the form. • (8.0) Modified the script on the form.		✓	✓	✓	✓
System:MoveContact • (7.5.1) For the rgCopyMove:TRadioGroup control, set ItemIndex to 1, and Text to Move the Contact to the New Account. Modified the script on the form. • (7.5.2) Modified the script on the form. • (7.5.4) Modified the script on the form. • (8.0) Modified the script on the form.	✓	✓		✓	✓
System:Opportunity Detail • (7.5.2) Modified the Width of controls on the form. Modified the TabOrder of controls on the form. For the ppeReseller:TPopupEdit control, set AllowClear to True, ReadOnly to False, and the OnChange event to ppeResellerChange. Removed the txtModifyDate:TEdit control. Modified the script on the form. • (7.5.4) Modified the script on the form.		✓		✓	
System:Preview Target List • (7.5.4)For the grdPreviewTargets:TDataGrid, set Sortable to False.				✓	
System:Select Contact(s) • (7.5.1) For the cboFilterBy:TComboBox control, set ItemIndex to -1 and Text to blank. For the lblFilterBy:TLabel control, set Width to 40. Modified the script on the form. • • (7.5.4) Modified the script on the form.	✓			✓	
System:SLX Report Manager View • (7.5.1) Modified the script on the form. • (7.5.2) Modified the script on the form. • (7.5.4) Modified the script on the form. • (8.0) Modified the Color of the TLabel2 control.	✓	✓		✓	✓
System:Ticket Detail • (7.5.1) Modified the Width of controls on the form. Modified the script on the form. • (7.5.2) Modified the script on the form.	✓	✓			
System:Update Multiple Opportunities • (7.5.2) Modified the Width of Label controls on the form. For the lveAcctManager:TLookupEdit control, set ReadOnlyEditor to True. Modified the script on the form. • (8.0) Modified the script on the form.		✓			✓
System:Update Targets • (7.5.4) For the grdTarget:TDataGrid control, set Sortable to False.				✓	
System:View History Attachments • (7.5.1) For the grdAttach:TDataGrid control, modified the SQL property. Modified the script on the form.	✓				
Ticket:Attachments • (7.5.1) For the grdAttach:TDataGrid control, modified the SQL property.	✓				

Form Name / Change	7.5.1	7.5.2	7.5.3	7.5.4	8.0
Ticket: Details • (7.5.1) Added the edtDescription: TEdit and edtResolution: TEdit controls. For the memDescription: TMemo control, set OnExitControl to memDescriptionExitControl. For the memResolution: TMemo control, set OnExitControl to memResolutionExitControl. Changed the Height and Width of controls on the form. Modified the script on the form.	✓				
Ticket: Ticket Activities • (7.5.1) Modified the Height and Width of controls on the form. Modified the script on the form. • (7.5.2) For the grdActivity: TDataGrid control, set DefaultRowHeight to 18 and Height to 375. For the memNotes: TMemo control, set Height to 375 and Width to 763. Modified the script on the form.	✓	✓			

Updated Global Scripts

See for more information on finding script changes.

Global Script Name	7.5.1	7.5.2	7.5.3	7.5.4	8.0
System: Global Campaign		✓			✓
System: Global SpeedSearch				✓	
System: Global System	✓			✓	

Updated VB Scripts

See for more information on finding script changes.

VB Script Name	7.5.1	7.5.2	7.5.3	7.5.4	8.0
Dashboard: Content Set Support		✓			✓
Dashboard: Export Support	✓				
Dashboard: Period Support		✓			
System: Attachment Support				✓	
System: ERP Link Base Common			✓		
System: Insert Opportunity Common					✓
System: Notes History Common	✓				
System: SLX Activity Support		✓			
System: Address Common		✓	✓		
System: SLX Crystal Report		✓	✓		
System: SLX Database Support			✓		
System: SLX Error Support			✓		
System: SLX Lead Historical Support				✓	
System: SLX Lead Info	✓				
System: SLX Lead Mappings		✓			✓
System: SLX Lead Support		✓			

VB Script Name	7.5.1	7.5.2	7.5.3	7.5.4	8.0
System: SLX Lookup Support			✓		
System: SLX Print Detail	✓				
System: SLX Report Condition Builder Controller	✓	✓			
System: SLX Report Conditions	✓	✓		✓	
System: SLX Report Controller	✓	✓			
System: SLX TimeZones		✓			
System: SLX Util			✓		
System: SLX_Common	✓	✓	✓		✓
System: SLX_Export_Group_To_Excel	✓				
System: SLX_Multi_Currency			✓		
System: SP_SalesProcessFunctions	✓	✓			✓
System: Spell Check			✓		
Ticket: Ticket Activity Support		✓			

Changes to Existing Groups

Group Name/Change	7.5.1	7.5.2	7.5.3	7.5.4	8.0
ACO Contact: Active Campaigns					✓
ACO Contact: All Contacts					✓
ACO Contact: Do Not Solicit					✓
ACO Contact: Duplicate Emails					✓
ACO Contact: Latest Contacts					✓
ACO Contact: My Activities for Today					✓
ACO Contact: My Contacts					✓
ACO Contact: My History					✓
ACO Contact: Status = Duplicate					✓
ACO Contact: Status = Purge					✓
ACO OPPORTUNITY: All Open					✓
ACO OPPORTUNITY: My Closed Opportunities					✓
ACO Opportunity: My Open Opportunities					✓
ACO OPPORTUNITY: My Pipeline					✓
ACO OPPORTUNITY: My Top Opportunities					✓
CONTRACT: Active Contracts • (7.5.2) Modified for localization.		✓			
CONTRACT: All Contracts • (7.5.2) Modified for localization.		✓			

Group Name/Change	7.5.1	7.5.2	7.5.3	7.5.4	8.0
CONTRACT: Pending Expirations • (7.5.2) Modified for localization.		✓			
HISTORY: All History					✓
HISTORY: Completed Activities					✓
HISTORY: My Completed Activities					✓
HISTORY: My Notes					✓
LITREQUEST: All Open					✓
PRODUCT: All Products • Modified for accounting integration.				✓	
PRODUCT: Available Products • Modified for accounting integration.				✓	
PRODUCT: Discontinued Products • Modified for accounting integration.				✓	
SALESORDER: All Sales Orders				✓	
SALESORDER: My Sales Orders				✓	
SALESORDER: Quotes				✓	
TICKETPROBLEMTYPE: All Standard Problems				✓	
TICKETSOLUTIONTYPE: All Standard Resolutions				✓	

Changes to Existing Menus and Toolbars

Name / Change	7.5.1	7.5.2	7.5.3	7.5.4	8.0
Menus System: Standard Menus • (7.5.1) Added Create New Group item under the CampaignNavMenu > Groups item. • (8.0) Removed Marketing Services item under the Standard Menu > Tools > Manage item.	✓				✓

Changes to Existing Reports

Name / Change	7.5.1	7.5.2	7.5.3	7.5.4	8.0
Account: Account Phone List - Sample • Removed table metadata that is no longer part of the Sage SalesLogix schema.	✓				
Contact: Contact Address Book - Sample • Removed table metadata that is no longer part of the Sage SalesLogix schema.	✓				
Contact: Contacts By Account - Sample • Removed table metadata that is no longer part of the Sage SalesLogix schema.	✓				
Opportunity: Potential Sales Opportunities - Sample • Removed table metadata that is no longer part of the Sage SalesLogix schema.	✓				

Changes to Existing Widgets

Widget Name	7.5.1	7.5.2	7.5.3	7.5.4	8.0
Dashboard Page System:My Dashboard					✓
Dashboard Page System:Sales					✓
Dashboard Widget System:Bar Chart					✓
Dashboard Widget System:Column Chart					✓
Dashboard Widget System:Default				✓	
Dashboard Widget System:Funnel Chart					✓
Dashboard Widget System:Group List • Updated to allow the grouplist to handle column widths expressed as strings and to call online help.				✓	✓
Dashboard Widget System:Line Chart					✓
Dashboard Widget System:Links					✓
Dashboard Widget System:Pie Chart					✓
Dashboard Widget System:Recently Viewed					✓
Dashboard Widget System:Todays Activities				✓	✓
Dashboard Widget System:Welcome				✓	✓

Chapter 4
Configuring Accounting Integration

Accounting integration allows you to share information between Sage SalesLogix and any accounting system that supports Sage CRM ERP Contract. Sage SalesLogix Web Client users can view open sales orders, customer invoices, customer payments, open invoices, and their corresponding details. This feature is not available to Sage SalesLogix Windows or Mobile Client users.

Sage SalesLogix supports the synchronization of information between Sage SalesLogix and supported accounting applications. However, depending on your accounting system, some of this information may not be available. Each accounting system determines the record types they will synchronize with Sage SalesLogix. Depending on your integration, you may be able to synchronize all or some of the following:

• Accounts/Trading Accounts

• Addresses

• Contacts

• Contracts/Invoices

• Opportunities/Sales Quotes

• Prospects/Customers

• Sales Orders

> Synchronization of information in real-time requires an internet connection. When using Accounting Integration on the Offline Web Client, data in real-time data views will be current as of the last time the Client connected with the host database.

Accounting Integration Features

When you integrate Sage SalesLogix with an accounting system, you can transfer information between systems so that your front-office and back-office systems work together and contain the same data. Users can:

• View current pricing and discounts in Sage SalesLogix with information pulled from the accounting system.

• Use price lists and products that originate in the accounting system.

• Submit sales orders from Sage SalesLogix to the accounting system and receive confirmation when the order has posted.

• View warehouse information and product availability from Sage SalesLogix.

• View order information such as price, discounts, shipping dates, and more in real-time.

Understanding Accounting Integration Prerequisites

Before configuring accounting integration, you MUST have upgraded your database by converting sales orders in the conversion to v8.0 (the conversion runs in the upgrade bundle). If the conversion is not run, mapping information for accounting integration may not work properly.

Installing Accounting Integration

To enable Accounting Integration, you must upgrade to v8.0 using the standard upgrade steps listed in Chapter 1, "Upgrade Workplan".

Understanding Customizations

If you have any customizations to the following form and plan to enable Accounting Integration, you must integrate your customizations to the IC version of the form. In v8.0, a Base and IC version of the following form are available. The Base version is used for non-Accounting Integration installations.

To integrate your customizations, use the Application Architect during the upgrade as detailed in Chapter 1, "Upgrade Workplan".

Configuring Accounting Integration

To configure Accounting Integration you must enable endpoints, start the service, import records, and configure users and roles.

> ⚠️ Configuring Accounting Integration modifies the connection to the Sage SalesLogix product table so it is no longer possible to add products to an open opportunity. If you are using these instructions to add Accounting Integration to an existing Sage SalesLogix 8.0 installation, make any necessary modifications to open opportunities first.

Running the Account Integration Phone and E-mail Update Utility

Before setting up your Accounting Integration endpoints, you must run the Account Phone Number and E-mail record update utility. This utility creates a record in the ERPPHONENUMBER or ERPEMAILADDRESS table for each phone number and email column in either of the ACCOUNT or CONTACT tables. Those records are mapped back to the original entity column where they originated.

> 📝 Windows Authentication must be configured for the administrative user before running the Account Integration Phone and E-mail Update utility. For detailed steps, see the "Using Windows Authentication" topic in the Administrator Help.

To run

1. On your Administrative Workstation, browse to the **ICLegacyPhoneEmailUpdater.exe** and **ICLegacyPhoneEmailUpdater.config** files.

   By default the configuration files are in …\Program Files\SalesLogix.

2. Double-click **ICLegacyPhoneEmailUpdater.config** to open the file in a text editor of your choice.

3. Scroll to the **<connectionStrings>** section and change the **Initial Catalog** value from **SALESLOGIX_EVAL** to your database alias.

   For example:

   <connectionStrings> <clear/> <add name="Default" connectionString="Provider=SLXOLEDB.1;Persist Security Info=True;Initial Catalog=MyDatabaseAlias;Data Source=localhost;Extended Properties="PORT=1706;LOG=ON;CASEINSENSITIVEFIND=ON;AUTOINCBATCHSIZE=1;SVRCERT=12345;ACTIVITYSECURITY=OFF;TIMEZONE=NONE""/> </connectionStrings>

4. Double-click **ICLegacyPhoneEmailUpdater.exe** to run the utility.
5. Click **Create ERP Phone Records** to create phone records.
6. Click **Create ERP E-mail Records** to create e-mail records.
7. Click the **Close** button.

Configuring Endpoints

Accounting Integration is enabled when the Sage SalesLogix endpoint and at least one target endpoint is configured and enabled. At least one endpoint must be active to share data between Sage SalesLogix and your accounting system.

To set

1. On the Web Client **Administration** menu, click **Integration Setup**.
2. In the **SalesLogix Feed** box, type the SData URL that Sage SalesLogix uses to share information between systems. Do one of the following:
    *   If your implementation does not use the default port (port 80), use format http://server:port/sdatavirtualdir/application/contract/operatingcompany/-/
    *   If your implementation uses port 80, omit the port number from the URL. The Sync Service automatically strips the default port from the URL which causes matching issues and errors. Avoid this by using format http://server/sdatavirutaldir/appliation/contract/operatingcompany/-/
    *   Set *application* to slx.
    *   Set *contract* to gcrm.
    *   Set *operatingcompany* to - (Sage SalesLogix does not have an operating company).
3. In the **User Name** box, type the user name for the Sage SalesLogix database that the accounting system connects to.
4. In the **Password** box, type the password for the user name.
5. In the **Display Name** box, type the display name for the feed.
6. Click **Save**.
7. If necessary, select the **Restrict account promotion to single accounting system** option to limit your integration to Sage SalesLogix and one accounting system.
8. Click the **Accounting Systems** tab to configure the feed details for your accounting system(s).
9. Click **Add**.

10. Set the following values in the **Setup Accounting Integration** dialog box.

Box	Description/Action
Accounting Feed	Type the SData URL to the accounting system. 1. Do one of the following: • For implementations that do not use the default port (port 80), use format http://server:port/sdatavirtualdir/application/contract/operatingcompany/-/ • For implementations that use port 80, omit the port number from the URL. The Sync Service automatically strips the default port from the URL which will cause matching issues and errors. Avoid this by using format http://server/sdatavirutaldir/appliation/contract/operatingcompany/-/ 2. Set *application* to your accounting application name. For example, Sage1000. 3. Set *contract* to gcrm. 4. Set *operatingcompany* to the operating company in your accounting application that you are integrating with.
User Name	Type the user name used to log on to the account system.
Password	Type the password for the corresponding user name.
Test Feed	Click to test and display the connection status.
Display Name	Type a display name for the accounting system. This name displays in the Web Client when users are linking records between Sage SalesLogix and the accounting system.
Enables Synchronization, Linking and Transacting	Select to enable this feed.
Filter By Resource	Select a resource type and click the Execute button to display schema differences in the grid. This feature compares the schema for the selected resource in the two integrated systems and returns any differences in string lengths. If string values do not match, use the Database Manager or other application to modify the schema in one of the systems.

5. Click **OK**.

Comparing and Configuring Schema Differences

Sage recommends comparing the database schema in Sage SalesLogix with your accounting application schema to ensure data integrity. You can compare data for various resources to determine if differences in string lengths exist between systems.

This feature compares the schema for the selected resource in the two integrated systems and returns any differences in string lengths.

To compare

1. On the Web Client **Administration** menu, click **Integration Setup**.

2. On the **Accounting Systems** tab, click **Edit** for the enabled accounting system.

3. In the **Setup Accounting Integration** dialog box, use the Schema Differences section to find differences between systems.

   a. In the **Filter By Resource** box, select the resource type.

   b. Click **Execute**.

   c. Review the differences in the grid.

   d. Note any information that may cause data integrity issues. For example, an accounting system field may only accept three characters when the same field in Sage SalesLogix accepts free text.

4. Repeat step 3 for all resource types.

5. Use the Database Manager or other application to modify the schema in one of the systems.

Configuring Matching Criteria

Use the Matching tab to set global default matching criteria for users creating links. These settings will apply to all users as they link accounts between Sage SalesLogix and your accounting system.

Sage SalesLogix has defined default matching criteria for accounts. To modify the criteria specific for your company, see the "Matching Tab" topic in the Web Client help.

Configuring the Accounting Integration Synchronization Service

The Accounting Integration Synchronization Service (SalesLogix SData Sync Service) allows data transfer between Sage SalesLogix and your accounting system. The service must be started and set to run as automatic.

To configure

1. Make sure you configure and enable endpoints as detailed in before starting the service.

2. Review the **Sage.SalesLogix.SyncEngineService.exe.config** file to ensure the connection is defined correctly at the end of the file. The connection information should match the **connection.config** deployed to your Web site.

   By default the configuration file is in …\Program Files\SalesLogix\Sage.SalesLogix.SyncEngineService.exe.config and the connection file is deployed to …\inetpub\wwwroot\slxclient

3. On the Administrative Workstation, open **Services**.

4. Right-click the **SalesLogix SData Sync Service**, and then click **Properties**.

   a. In the **Startup type** box, select **Automatic**.

   b. Set the service to log on and run as the WebDLL user.

      Ensure you have enabled Windows Authentication for the WebDLL user.

   c. Click **Start**.

5. Click **OK**.

Exporting and Merging Records from your Accounting System

After configuring the integration endpoints and starting the synchronization service, you can export records from your accounting system to Sage SalesLogix. Any records imported into Sage SalesLogix will be linked to the corresponding record in your accounting system. After exporting records, you should check for duplicate records.

The Check for Duplicates wizard compares records in Sage SalesLogix to determine if your data contains duplicates. The wizard creates a job containing possible duplicate data. You can use the job results to merge or remove duplicates.

Sage recommends that you establish a weekly schedule to check for duplicates in your system.

To export and merge

1. Use the export tool provided by your accounting system to export the appropriate data into Sage SalesLogix.
2. On the Web Client **Tools** menu, click **Check for Duplicates**.
3. On the **Select a Source** screen:
    a. In the **Select type of Job drop**-down list, select the entity type for which you want to check for duplicates.
    b. In the **Group** drop-down list, select the group of records within the selected entity type.
    c. Click **Next**.
4. On the **Search Options** screen, select the filters you want the wizard to use when matching data.

    If you select a value, the data in both records must be an exact match to be considered a duplicate.
5. (Optional) Click **Advanced Match** to set additional match options.
6. Click **Next**.
7. On the **Review** screen, verify your selections for this job.
8. Click **Submit**.
9. Click the **Job Number** link to open the Check for Duplicates History Detail view and view any potential duplicates.
10. On the **Potential Duplicates** tab, click the **Resolve** link.
11. In the **Duplicate Search Results** section:
    *   Click the **Merge** link for the account with which you want to merge this record. Then, use the **Merge Data** dialog box to select the data you want to keep.
    *   Click **Not a Duplicate** if the record is not a duplicate and should be left as a separate record.
12. Return to the **Check for Duplicates History Detail** view and merge or mark records as not a duplicate for all records in the job.

Configuring User Roles

By default, users created in the Web Client are granted the Standard User role which allows access to accounts, contacts, opportunities, and other Sage SalesLogix records. When enabling Accounting Integration, you should assess the roles required for each user depending on the tasks they will perform.

Review the "What are Roles and Secured Actions?" topic in the Web Client help to determine what access is required for each of your Sage SalesLogix Accounting Integration users and use the Roles view to assign and remove your user's roles as appropriate.