Applying Service Pack 2 for Sage SalesLogix Version 7.5

Version 7.5.2

Developed by Sage SalesLogix User Assistance

Applying Service Pack 2 for Sage SalesLogix Version 7.5

Documentation Comments

This documentation was developed by Sage SalesLogix User Assistance. For content revisions, questions, or comments, contact the Sage SalesLogix writers at saleslogix.techpubs@sage.com.

Version

Version 7.5.2
2009

Technical Support

Technical Support is available to customers with support contracts directly from Sage 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. Customers with a valid technical support contract and a Web access code can request technical support electronically on the Sage SalesLogix SupportOnline/Sage Software Online Support and Services Web site.

Sage SalesLogix Web site: www.sagecrmsolutions.com
Sage SalesLogix SupportOnline/Sage Software Online Support and Services Web site: http://support.saleslogix.com

Revisions to this book are posted on the Sage SalesLogix SupportOnline/Sage Software Online Support and Services Web site. Check this site regularly for current Sage SalesLogix product documentation.

Contents

Chapter 1
Upgrade Workplan

Use this workplan to upgrade the Sage SalesLogix Network and Web components to Sage SalesLogix version 7.5 Service Pack 2. Follow the tasks in order. If a task does not apply, disregard it and move on to the next.

The "Required for" column in this workplan indicates you must complete the task for:

- **All** - all Sage SalesLogix installations.
- **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 to version 7.5.2, 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.
	If you have not upgraded to Sage SalesLogix v7.5, refer to the appropriate upgrade document on the SupportOnline\Sage Software Online Support and Services Web site (http://support.saleslogix.com) for instructions.

All	**2** Install this release in a test environment before installing on your production database. Refer to SupportOnline\Sage Software Online Support and Services Knowledge Base 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.

Required for:	Task Description
All	**3** Grant admin rights to the person installing the service pack. Disconnected Web Client users can upgrade without admin rights on their machines. Other users require admin rights when installing Sage SalesLogix.
All	**4** Install or activate .Net Framework v3.5 SP1. This release requires .Net Framework version 3.5 SP1 on all computers where the Application Architect and/or Sage SalesLogix Personal Web Server are installed. This includes computers running Intellisync for SalesLogix (for example, disconnected Web Clients and Sage SalesLogix Remote Clients). **Important** Before installing the service pack, install or activate .Net Framework. Some Windows versions (such as Windows 2008 R2) have .Net Framework v3.5 SP1 installed, but not activated.
Protect Your Customizations	
Customized Network	**5** 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.
Customized Network	**6** 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.
Customized Web v7.5.0	**7** Document the values for the Mashup Name and Result Name properties for all Timeline Visualization controls on each of your custom Web forms. **Note** This task is required when upgrading from v7.5.0 to v7.5.2. This task is not required if you are upgrading from v7.5.1 to v7.5.2. 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 40 describes how to apply the changes after the upgrade.
Customized Web	**8** 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. Then, Task 41 describes how to apply the changes after the upgrade. **Important** You must record your customizations before upgrading to v7.5.2. When you open the Application Architect after upgrading, all Hibernate.xml files are deleted.

Required for:	Task Description
Customized Web	**9** Document or save any customizations made in the web.config file(s). All web.config files are overwritten with the service pack upgrade. Any customizations to these files are not included in your customization bundle. A web.config file exists for each portal. With an IIS installation, the file is located in: • (XP/2000/2003) \\Documents and Settings\All Users\Application Data\SalesLogix\Sync\PortalDeployments*[portal name]*. • (Vista/7/2008) \\ProgramData\SalesLogix\Sync\PortalDeployments*[portal name]*. Before you upgrade, use the Application Architect to note customizations in all web.config files.
Customized Web with Hot Fixes	**10** Create a new project workspace and add the hot fixes currently installed on your system. This workspace will be used to create a bundle of your Web customizations. **Note** If you have not customized your Web implementation, disregard this task and proceed to Task 14. If you customized your Web implementation, but do not have any hot fixes installed, proceed to Task 11. After creating a new project workspace, restore the project backup into the workspace, and apply your hot fixes. **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. The project backup files are included in the service pack download files. **Note** Additional project backups are available for upgrades from Release Candidate (RC) product versions. Disregard the v7.5 SP2 RC project backups unless you are currently using a Release Candidate version. 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 currently installed in your production environment to the new project. 4 Proceed to Task 12.

Required for:	Task Description
Customized Web Without Hot Fixes	**11** 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 14. If you customized your Web implementation and have hot fixes installed, proceed to Task 12. 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. The project backup files are included in the service pack download files. **Note** Additional project backups are available for upgrades from Release Candidate (RC) product versions. Disregard the v7.5 SP2 RC project backups unless you are currently using a Release Candidate version. 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 12.

Required for:	Task Description
Customized Web	**12** Create a bundle of your Web customizations. **Note** If you have not customized your Web implementation, disregard this task and proceed to Task 14. **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 on your local computer (Task 11). • If your custom installation includes hot fixes, select **Existing Project**, and then select the project you created in Task 10. 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 Ensure the **Exclude the Following Files** check box is selected. Add **mnuSpeedSearch.menu.xml** to the list. Then, add any additional files you do not want to include in your bundle (even if they have been added or changed). By default, various Sage files are listed. For example, the Sage.*.dll excludes Sage assemblies from the bundle. The list supports wildcards using the asterisk character. Exclusion matching is performed against the URL, not just the filename, so folders may also be specified. 6 Click **OK**. The new manifest opens in the Application Architect. 7 On the New Manifest **Properties** tab, type a name in the **Name** box. For example, Customizations. 8 Save the manifest. 9 Expand **Bundle Model**, right-click the new manifest, and then click **Create Bundle**. 10 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.
Customized Web	**13** Save any customized Web Help files to a location outside the VFS. In this release, the Web Help has been converted from .aspx files to .htm. In future releases your customizations will be captured in the bundle process. However, for this release you must recreate your customizations in the new help topics. See the "Upgrading Customized Help" topic in the Developer Tips Help for details.
Prepare Your Sage SalesLogix Database	
Microsoft SQL Server	**14** 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.
All	**15** Make sure all users have logged off Sage SalesLogix.
Remotes	**16** Instruct all Remote users and Offices to run a final synchronization cycle.

Required for:	Task Description
Remotes	**17** Execute a final cycle of all sync servers. The final sync cycle must run successfully and send out all pending transactions.
Remotes	**18** Stop the Sync Service(s). If using a third-party scheduling package, deactivate the Synchronization Server tasks.
All	**19** 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	**20** Back up your production database. Verify the backup is successful and complete.
Upgrade your Main Office	
All	**21** Run the SLX_v75_SP2 installation on every computer at the main office where a Sage SalesLogix component is installed. **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. Depending on the installation method you choose, service pack files and bundles may not be extracted to your local computer. • If you install the service pack from a temporary folder without extracting all the contents to your local computer, the bundles remain in the EXE file within the service pack .zip file and you must extract the bundles from the EXE when you are ready to install them. • If you choose to extract the files and then install the service pack from your local computer, the bundles are included in the extraction process and can be found on your local computer when you are ready to install them. **To run** 1 Extract the contents of the **SLX_v75_SP2.exe** file to a temporary folder. Before upgrading your Synchronization Server, ensure the Synchronization Service (SLXSyncService) is stopped. In addition, shut down any instances of Synchronization Server in Task Manager. 2 Navigate to the folder where you extracted the service pack files and double-click **SLX_v75_SP2.exe**.

Required for:	Task Description
All	**Task 21 - continued** 3 On the **SalesLogix v7.5 Service Pack 2** screen, select your installation method. Depending on the installation method you choose, service pack files and bundles may not be extracted to your local computer. Ensure you select the appropriate installation method for your upgrade. • The **Install** option extracts the service pack files to a temporary folder and removes them once the installation is complete. If you install the service pack from a temporary folder without extracting all the contents to your local computer, the bundles remain in the EXE file within the service pack .zip file and you must extract the bundles from the EXE when you are ready to install them. • The **Extract and Install** option extracts the service pack files to a location you specify. The files are not removed once the installation is complete. If you choose to extract the files and then install the service pack from your local computer, the bundles are included in the extraction process and can be found on your local computer when you are ready to install them. Selecting the Extract and Install the Service Pack option allows you to distribute a smaller upgrade patch to your Sage SalesLogix Network and Remote users. This process is detailed in Task 30 and Task 31. 4 Click **Next**. 5 On the **Welcome** screen, verify all components installed on this computer appear in the list, and then click **Install**. The Update Network Images button is available on the Welcome screen. If you created automated installations of the Sage SalesLogix Client, you can upgrade the installations using this option (you must browse to the .msi file and click Update). Automated installations can be used to install the Sage SalesLogix Client for new users. When properly updated, this installation contains the initial version of the Sage SalesLogix Client and all service packs and hot fixes that have been applied to your system. 6 When the installation indicates a successful upgrade, click **Finished**. The upgrade is successful when the Status column displays Complete. 7 Restart your computer after the installation to ensure that all changes take effect. You may automatically receive a prompt to restart.
All	**22** Start the Administrator.
All	**23** 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.

Required for:	Task Description
Install the Service Pack Bundle	
All	**24** Use the Administrator to apply the service pack bundle named SalesLogix v7.5 Service Pack 2.sxb. See the "Installing a Bundle" topic in the Administrator Help for instructions. If you extracted the service pack files to your local computer (Task 21), the bundle is located in the folder you specified. If you installed the service pack without extracting the files, you must extract the bundle from the SLX_v75_SP2.exe before installation. 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** 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** You must log on to the conversion utility as admin. **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. **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.

Required for:	Task Description

All	**Task 24 - 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.
	
	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.
	
	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	**25** Instruct your Oracle DBA to run the CreateOracleViews_75_SP2.sql script. The CreateOracleViews_75_SP2 script updates the Campaign Targets view. Running this script allows you to create the necessary Sage SalesLogix view without extra permissions for the sysdba user. **Note** The CreateOracleViews_75_SP2.sql script must be run as the sys user. **To run** 1 Open Oracle SQL *Plus Worksheet or SQL Developer. 2 On the **File** menu, click **Open**. 3 Browse to the **CreateOracleViews_75_SP2.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	**26** 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.
Customized Network	**27** 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	**28** 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	**29** Restart the Sync Service(s).
Upgrade Network and Remote Users/Offices	
All	**30** Upgrade each of your Sage SalesLogix Network user's computers using the SLX_v75_SP2 installation or the appropriate .MSP file. You can upgrade Network users by distributing an .MSP file rather than the entire service pack patch. This file is smaller and upgrades only the necessary Sage SalesLogix Client pieces. The .MSP files are located in the folder you specified when extracting the service pack files (see Task 21). After receiving the file, instruct your Network users to double-click **SLX_v75_SP2.exe** or **SalesLogix Client 7.5 SP2.msp** as appropriate.

Required for:	Task Description
Remotes	**31** Distribute Client installations or .MSP files to your Remote users and Offices. You may want to distribute via e-mail or copy the installations to a shared network drive. You can upgrade Remote users and Offices by distributing an .MSP file rather than the entire service pack patch. This file is smaller and upgrades only the necessary pieces. The .MSP files are located in the folder you specified when extracting the service pack files (see Task 21). Distribute the following .MSP files as appropriate: • For Remote users: **SalesLogix Client 7.5 SP2.msp** • For Remote Offices: **SalesLogix Remote Office 7.5 SP2.msp** • For disconnected Web Client users: **SalesLogix Disconnected Web Client 7.5 SP2.msp**
Remotes	**32** Direct Remote users and Offices to upgrade using the SLX_v75_SP2 installation or the appropriate .MSP file. Before upgrading a disconnected Web Client, instruct users to close the Sage SalesLogix Personal Web Server. After upgrading, restart the server. • To stop the server, right-click the **Sage SalesLogix Personal Web Server** icon in the application tray, and then click **Exit**. • To start the server, click **Start**, point to **Programs**, point to **Sage SalesLogix**, and then click **SalesLogix Web Server**. **Upgrading Without Admin Rights on Windows XP and 2003** If you want to upgrade disconnected Web Client users without granting the user admin rights, use the following steps. **To upgrade** 1 Stop the Sage SalesLogix Personal Web Server. 2 Stop the following services: • SalesLogix Server • SalesLogix SpeedSearch Service • SalesLogix System 3 Upgrade using the following command: Msiexec /p <path to msp> /qn REINSTALL=ALL REINSTALLMODE=omus NOADMINACNT=TRUE Change <path to msp> to the location of the SalesLogix Disconnected Web Client 7.5 SP2.msp on the local machine. Use the format C:\SLX\ 4 Restart your computer.
Remotes	**33** 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.

Required for:	Task Description
Upgrade Your Web Workspace	
All Web	**34** Create a new project workspace. After creating a new project workspace, you can restore the v7.5.2 project model from backup into the workspace. This creates a v7.5.2 project workspace. **To create** 1 On the Application Architect **View** menu, click **Project Workspace Manager**. 2 Right-click in the **Project Workspaces** window, and then click **Add**. The Add Project Workspace dialog box opens. 3 In the **Name** box, type a name for the project workspace. 4 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. 5 In the **Description** box, type a description or additional information about the new project. 6 Ensure the **Export Files Upon Creation** check box is not selected. 7 Click **Create**.
All Web	**35** Restore the Sage SalesLogix v7.5 SP2 project backup to the new project workspace (created in Task 34). 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 restore** 1 Locate the **Sage SalesLogix v7.5 SP2 Project Backup.zip** file. If you selected the Extract and Install option when running the service pack installation, the default location for project backup files is: • (XP/2000/2003) \\Documents and Settings\username\Local Settings\Temp\SalesLogix\Project Backups. • (Vista/7/2008) \\Users\username\AppData\Local\Temp\SalesLogix\Project Backups. **Note** Additional project backups are available for use when creating a customization bundle (Tasks 10 and 11). Use the Sage SalesLogix v7.5 SP2 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 project workspace you created in Task 34. 5 Right-click the project, and then click **Restore Project**. 6 In the **Select Project Backup File to Restore From** dialog box, browse to the Sage SalesLogix v7.5.2 project backup on your local machine (step 2). 7 Click **OK**.
All Web	**36** 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 53.

Required for:	Task Description
Reapply Your Web Customizations	
Customized Web	**37** Use the BundleDiff utility to preview how Sage changes impact your customizations. You can use Sage differences bundles with the BundleDiff utility to find differences between two bundle manifests. Use this information to determine how you want to merge your customizations with the Sage changes before you apply your custom bundle. The differences bundles are located in the Project Differences folder. The BundleDiff utility is installed in the \\Program Files\SalesLogix folder. **BundleDiff Parameters** • **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. 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.

Required for:	Task Description
Customized Web	**38** Use 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 12) to the new v7.5.2 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 7.5.2 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 (this may depend on your differencing tool). **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 12 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.

Required for:	Task Description
Customized Web	**Task 38 - continued** 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. The behavior at install for the duplicate items is controlled by the selected install action. 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 (.resx files). Your differencing tool will display them twice; once for the main file, and once for the .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. 12 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. 13 Click **Next**. 14 Click **Finish**.
Customized Web	**39** If necessary, reapply your custom schema to the database. If you are restoring a project to the same database that your customizations were created on, 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.

Required for:	Task Description
Customized Web v7.5.0	**40** Update Timeline Visualization controls on your custom forms. **Note** This task is required when upgrading from v7.5.0 to v7.5.2. This task is not required if you are upgrading from v7.5.1 to v7.5.2. **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 7 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.
Customized Web	**41** 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 8 to update the template.
Customized Web	**42** Reapply your customizations to the web.config file(s). Use the information recorded in Task 9 to reapply your customizations. **Important** Make sure you integrate your customizations with the updated web.config file. Do not overwrite the Sage version with your customized version.
Customized Web	**43** 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	**44** 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 smarts 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
	`protected void cmdOK_Click(object sender, EventArgs e)`
	`{`
	`// ... save/update logic ...`
	`// Close dialog`
	`DialogService.CloseEventHappened(sender, e);`
	`// Refresh the page`
	`Refresh();`
	`}`
	Method 2: Wire up a Refresh event handler (also newly added to SmartPart class)
	`protected override void OnWireEventHandlers()`
	`{`
	`base.OnWireEventHandlers();`
	`cmdOK.Click += cmdOK_ClickAction;`
	`cmdOK.Click += DialogService.CloseEventHappened;`
	`cmdOK.Click += Refresh;`
	`cmdCancel.Click += DialogService.CloseEventHappened;`
	`}`

Required for:	Task Description

Customized Web	**45** 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;`

Customized Web	**46** 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
Customized Web	**47** 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**.
Customized Web	**48** 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** Alternatively, you may want to replace your SData consumption code with calls to the new SData Core Client Libraries. The SData Core Client Libraries simplify the coding effort to consume SData feeds. They are available for Technical Preview on the Sage SalesLogix Community Web site. Refer to http://sdata.sage.com for information on the Sage Data specification.

Required for:	Task Description

Customized Web	**49** 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:
	
	`public IEnumerable<ScriptDescriptor> GetScriptDescriptors()`
	`yield break;`
	`}`
	`public IEnumerable<ScriptReference> GetScriptReferences()`
	`{`
	`yield return new`
	`ScriptReference("~/SmartParts/Opportunity/AddOpportunityProduct.js");`
	`}`
	
	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:
	
	`protected override void OnPreRender(EventArgs e)`
	`{`
	`if (DesignMode == false)`
	`if (ScriptManager.GetCurrent(Page) != null)`
	`(ScriptManager.GetCurrent(Page).RegisterScriptControl(this);`
	`}`
	
	3 Ensure that Sys.Application.notifyScriptsLoaded is called at the end of the referenced script, for example:
	
	`"if (typeof(Sys) !== 'undefined') Sys.Application.notifyScriptLoaded();`
	
	The script will now be correctly referenced when the control is added to the page during a partial post back.

Required for:	Task Description

Customized Web	**50** 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.
	• 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
	If you have customized the MailMerge.vbs using the following CreateObject statements, update these objects to the new class name.
	• 7.5.2 MailMerge.vbs
	  • CreateObject("SLXMMEngine.MailMergeEngine") change to CreateObject("SLXMMEngineW.MailMergeEngine")
	  • CreateObject("SLXMMGUI.EditMergedDocs") change to CreateObject("SLXMMGUIW.EditMergedDocs")
	  • CreateObject("SLXMMGUI.fFaxOptions") change to CreateObject("SLXMMGUIW.fFaxOptions")
	  • CreateObject("SLXMMGUI.SelectAddressType") change to CreateObject("SLXMMGUIW.SelectAddressType")
	  • CreateObject("SLXMMGUI.SelectPrinter") change to CreateObject("SLXMMGUIW.SelectPrinter")

Customized Web	**51** Update your custom fax provider. As noted in the Task 50, 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 the SLXFaxW fax provider.
Customized Web	**52** 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 Custom Form" 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.

Required for:	Task Description
Upgrade Web Reporting	
Web v7.5.0	**53** Upgrade Web Reporting. **Note** This task is required when upgrading from v7.5.0 to v7.5.2. This task is not required if you are upgrading from v7.5.1 to v7.5.2. When upgrading from v7.5.1, your existing Web Reporting portal is updated when you build and deploy in Task 55. 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. **Note** If you want to add Web Reporting to a Web Remote Office, you must install the Web Reporting Server at the Remote Office using the Sage SalesLogix v7.5 DVD. Then upgrade the server using the following steps. **To upgrade** 1 Run the SLX_v75_SP2 installation on your Web Reporting Server. 2 Open the Application Architect Deployment Explorer. 3 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**. 4 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. 5 Click **Save**. 6 Click **Deploy**.

Required for:	Task Description
Web v7.5.0	**Task 53 - continued** 7 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**. 8 In the Advanced Options section, click **Web Dependencies**. 9 In the **Manage Custom Service Host Entities** dialog box, click **Add**. The Service Host Entry Editor dialog box opens. 10 In the **Entity Name** box, type *slxwebrpt*. 11 In the **Service URL** box, type the URL for the Web Reporting portal. Use the format: http(s)://server_name:port_number/slxwebrpt. 12 Click **OK**. Then, click **Done**. 13 Save and deploy the Web site. 14 Reset IIS.
Build and Deploy your Web Portals	
All Web	**54** Delete the ReferenceAssemblies folder. Before building and deploying your Web portal(s), you must delete the ReferenceAssemblies folder on the Application Architect computer. If this folder exists when building the portal(s), you may see errors. By default, the ReferenceAssemblies folder is located in \\Program Files\SalesLogix. To verify the folder location, check the Assemblies Path information in the Application Architect. **To find the folder location** 1 On the Application Architect **Tools** menu, click **Build Settings**. 2 In the **Assemblies Path** box, note the location of the ReferenceAssemblies folder. 3 Using the location in step 2, delete the ReferenceAssemblies folder.
All Web	**55** 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 a Portal" topics in the Application Architect Help for instructions. **Important** Ensure you complete Task 54 before you build and deploy your Web portal(s).
Remotes	**56** 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.

Required for:	Task Description
Remotes	**57** Instruct disconnected 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. **Disconnected Web Client Users and Web Reporting** Changes in this release allow disconnected 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). Disconnected Web Client users must have the "Use ActiveReporting" option enabled in the Web Client to use this feature. **Note** Active Reporting requires Internet Explorer. The ActiveX components used in this feature will not run on Firefox. If Active Reporting is disabled, disconnected 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 Disconnected Web Client users upgrade their Sage SalesLogix installation (Task 33). 2 New remote portals containing Active Reporting components are built and deployed at the main office (Task 55). 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 56). 4 Disconnected Web Client users run a sync cycle to receive the updated portal (Task 57). 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 Disconnected Web Client users can use reporting with data from their local machine.
All Web	**58** 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
Upgrade Intellisync for SalesLogix	
Intellisync for SalesLogix	**59** Understand the Intellisync for SalesLogix upgrade. In this release the upgrade process updates Intellisync without additional installation or configuration steps. **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 21. • Network Clients receive updates when you complete Task 30. • Remote Clients and Offices receive updates when you complete Task 32. **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 55. • Network Clients receive updates when you complete Task 30. • Remote Clients and Offices receive updates when you complete Task 32. • Web Clients receive updates after you deploy the updated SlxIntellisync portal (Task 55) and they run a synchronization cycle to receive it (Task 57). 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**.

Required for:	Task Description
Intellisync for SalesLogix	**Task 59 - continued** **To Re-Sync** 1 With Intellisync open, click **Sync Now**. 2 On the **Confirm Edits** screen, click **Re-Sync**. 
Perform Post-Upgrade Tasks	
Web	**60** If necessary, enable Windows Authentication for Firefox. The following instructions configure Windows Authentication for Web Client users accessing Sage SalesLogix with Firefox. **To enable** 1 Open Firefox. 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	**61** 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. • Example Configuration Settings: `<httpCompression directory="%SystemDrive%\inetpub\temp\IIS Temporary Compressed Files" maxDiskSpaceUsage="500" noCompressionForHttp10="false" noCompressionForProxies="false" > <scheme name="gzip" dll="%Windir%\system32\inetsrv\gzip.dll" dynamicCompressionLevel="10" staticCompressionLevel="10"/> <scheme name="deflate" dll="%Windir%\system32\inetsrv\gzip.dll" dynamicCompressionLevel="10" staticCompressionLevel="10" /> <staticTypes> <add mimeType="text/*" enabled="true" /> <add mimeType="message/*" enabled="true" /> <add mimeType="application/*" enabled="true" /> <add mimeType="*/*" enabled="false" /> </staticTypes> <dynamicTypes> <add mimeType="text/*" enabled="true" /> <add mimeType="message/*" enabled="true" /> <add mimeType="application/*" enabled="true" /> <add mimeType="*/*" enabled="false" /> </dynamicTypes> </httpCompression>`

Required for:	Task Description
Web	**62** Disable HTTP compression for mail merge and SendSLX. The components used by mail merge and SendSLX use zip compression to reduce the size of mail merge requests and responses. When HTTP compression is enabled in IIS this can conflict with mail merge and SendSLX when dynamic compression is applied to ASHX (IIS6) or "*/*" (IIS7). In addition, if third-party compression is used, both mail merge and SendSLX will fail unless compression is disabled. When HTTP compression conflicts occur you will see the following HTTP error: "The result for 1 queries was expected, but no data was returned. There may have been more than 1 HTTP data request active at the same time. The SQL syntax may also be incorrect." Use one of the following sections to disable compression for your IIS version. **To disable IIS6 compression (Windows 2003)** 1 Download and install Internet Information Services (IIS) 6.0 Resource Kit Tools. For download information, go to www.microsoft.com and search on 6.0 resource kit tools. 2 Stop the IIS Admin Service using the command: NET STOP "IIS Admin Service". 3 On the **Start** menu, point to **Programs**, point to **IIS Resources**, point to **Metabase Explorer**, and then click **Metabase Explorer**. 4 Locate and select the node that represents the SalesLogix Web site (SlxClient): Expand your server, expand **LM**, expand **W3SVC**, expand the Web site ID, expand **ROOT**, and select your Web site. 5 Right-click the SalesLogix Web site, point to **New**, click **Key**, and then type **SLXMailMergeServer.ashx** as the key name. 6 Right-click the new SLXMailMergeServer.ashx key, point to **New** and then click **DWORD Record**. The New Record dialog box opens. 7 In the **Record Name or Indentifier** drop-down list, select **DoDynamicCompression**, and then click **OK**. 8 Right-click the **DoDynamicCompression** record in the grid, and click **Properties**. The DoDynamicCompression Properties dialog box opens. 9 Click the **General** tab. a In the **User Type** drop-down list, select **File**. b In the **Attributes** section, select **Inheritable**. c Click **OK**. 10 Reset IIS. **To disable IIS7 compression (Windows Vista/2008)** 1 Stop the IIS Admin Service using the command: NET STOP "IIS Admin Service". 2 In Notepad, open the applicationHost.config file from %SystemRoot%\system32\inetsrv\config\.

Required for:	Task Description

Web	**Task 62 - continued**
	
	3 In the httpCompression section, add the application/x-zip-compressed mime type to the dynamicTypes section and set enabled to false.
	
	`<httpCompression directory="%SystemDrive%\inetpub\temp\IIS Temporary Compressed Files">`
	`<scheme name="gzip" dll="%Windir%\system32\inetsrv\gzip.dll" />`
	`<dynamicTypes>`
	`<add mimeType="text/*" enabled="true" />`
	`<add mimeType="message/*" enabled="true" />`
	`<add mimeType="application/x-zip-compressed" enabled="false" />`
	`<add mimeType="*/*" enabled="true" />`
	`</dynamicTypes>`
	`<staticTypes>`
	`<add mimeType="text/*" enabled="true" />`
	`<add mimeType="message/*" enabled="true" />`
	`<add mimeType="application/x-javascript" enabled="true" />`
	`<add mimeType="*/*" enabled="true" />`
	`</staticTypes>`
	`</httpCompression>`
	
	4 Save the applicationHost.config file.
	
	5 Reset IIS.

All	**63** 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.
All	**64** 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. **To create** 1 Using Notepad, create a blank UDL file named “test.udl”. 2 Save the UDL file to your C: drive. 3 Run the following command to open the blank UDL file: C:\Windows\syswow64\rundll32.exe "C:\Program Files (x86)\Common Files\System\Ole DB\oledb32.dll",OpenDSLFile C:\test.udl 4 With the blank UDL open, open the Data Link Manager. To open the Data Link Manager, open a Sage SalesLogix application and click the ellipsis button in the Log on to box. 5 In the Data Link Manager, create a new database connection and make sure you select the SalesLogix OLE DB Provider on the Provider tab. 6 Save the connection. 7 Move the UDL file to all computers that require a connection to Sage SalesLogix. By default, the file should be stored in C:

You have completed your upgrade. See the “What’s New in this Release” topic in the online Help systems for an explanation of the new features.

If necessary, in the location where you extracted the service pack files (Task 21), delete the extracted files. These files are automatically deleted if you selected the Install the Service Pack option.

Chapter 2
Changes in this Release

Version 7.5.2 is a cumulative release which includes all changes released since version 7.5.0.

Note Any change that impacts customizations 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 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

- .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.
- [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.
- You can choose how Send SLX handles attachments. See Task 24 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 50 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.

- Disconnected Web Client users do not need admin rights to upgrade their machines. See Task 32 for details.
- Disconnected Web Client users can generate reports locally against their remote database. See Task 57 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.

    Note Sales Orders in the Network Client are unchanged.

Performance Enhancements

The Web Client has improved performance in the following areas:

- 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.
- [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 47 for more information.
- 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.
- [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 44 for more information.
- 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.

  Note 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.

- 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.
- [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 43. 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 48.
- [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 49 to upgrade.
- 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).
- [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.
- 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.
- Disconnected Web Client users can subscribe/unsubscribe to accounts using the What's New view.
- Mail Merge is available for disconnected 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.

Note 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).

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 for v7.5.2 on SupportOnline/Sage Online Support and Services 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. 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 Fields

Table Name	Field Name	Purpose	7.5.1	7.5.2
ACTIVITY	ATTACHMENTCOUNT	Stores the number of attachments for the activity item.		✓
ATTACHMENT	SALESORDERID	Assigning a SalesOrderId associates this attachment to the SalesOrder.		✓
CALCULATEDFIELDDATA	SORTORDER	Added for performance improvements. For Sage use only.		✓
HISTORY	ATTACHMENTCOUNT	Stores the number of attachments for the history item.		✓
SALESORDER	ACCOUNTMANAGERID	Stores the assigned seccodeid.		✓

Table Name	Field Name	Purpose	7.5.1	7.5.2
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	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	GRANDTOTAL	Stores the grand total of the sales order. This field is used for groups and reporting.		✓
SALESORDER	REQUESTEDBY	Stores the contact ID that requested the sales order.		✓
SALESORDER	SECCODEID	Stores the "owner" (Account seccodeid) of the sales order.		✓
SALESORDERITEMS	CALCULATEDPRICE	Stores the items price * discount. The main purpose is for binding.		✓
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.		✓
VIRTUALFILESYSTEM	ISCOMPRESSED	Indicates whether the data is compressed (zip) in the record.	✓	

New Forms

Form Name	7.5.1	7.5.2
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
SALESORDER:All Sales Orders		✓
SALESORDER:My Sales Orders		✓
SALESORDER:Quotes		✓

Changes to Existing Main Views

Main View Name / Change	7.5.1	7.5.2
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
Account:Attachments • For the grdAttach:TDataGrid control, modified the SQL property.	✓	
Account:Contacts • For the frmContacts:TAXForm, set Width to 257. For the grdContacts:TDataGrid control, set Width to 245. Modified the script on the form.		✓
Campaign:Stages Tasks • For the grdStage Task:TDataGrid control, modified the Height and Width. 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.	✓	✓
Contact:Attachments • For the grdAttach:TDataGrid control, modified the SQL property.	✓	
Contact:Opportunities • (7.5.1.10) For the grdOppContacts:TDatGrid control, removed the OnCompareNotes property. Modified the script on the form.		✓
Contact:Tickets • Modified the Width of controls on the form. Modified the script on the form.		✓
Contract:Attachments • For the grdAttach:TDataGrid control, modified the SQL property.	✓	
Contract:Covered Assets • Modified the script on the form.	✓	
Contract:Tickets • Modified the script on the form.	✓	

Form Name / Change	7.5.1	7.5.2
Defect:Attachments • For the grdAttach:TDataGrid control, modified the SQL property.	✓	
Defect:Tickets • Modified the Width of controls on the form. For the txtSubject:TEdit control, set Text to blank.		✓
Lead:Attachments • For the grdAttach:TDataGrid control, modified the SQL property.	✓	
Opportunity:Attachments • For the grdAttach:TDataGrid control, modified the SQL property.	✓	
Opportunity:Contacts • Modified the script on the form.	✓	
Opportunity:Order Details • For frmOrderDetails:TAXForm, set Caption to Sales Orders. For the grdSalesOrders:TDataGrid, set Height to 379, and Width to 285.		✓
Opportunity:Sales Process • Modified the script on the form.	✓	
RMA:Attachments • For the grdAttach:TDataGrid control, modified the SQL property.	✓	
RMA:Return Details • Modified the script on the form.	✓	
Sales Dashboard:Opportunity Detail Analysis View • 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:Open Opportunities Analysis View • 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:Pipeline Status Analysis View • 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 Potential Analysis View • 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.	✓	✓
System:Add Edit Address • Modified the script on the form.		✓
System:Add Edit Campaign Stage • For the txtDescription:TEdit control, set MaxLength to 64.	✓	
System:Add Edit Response • Modified the script on the form.	✓	

Form Name / Change	7.5.1	7.5.2
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.	✓	✓
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 • Modified the script on the form.		✓
System:Add Targets From Group • Modified the script on the form.	✓	
System:Campaign Detail • Modified the Width of Label controls on the form. Modified the script on the form.		✓
System:Close Opportunity • Modified the Width of controls on the form. Modified the script on the form.	✓	
System:Create Group From Targets • 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 • Modified the script on the form.		✓
System:Email - Send Ticket Information • 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.	✓	✓
System:Insert Campaign • Modified the Width of controls on the form. Modified the script on the form.	✓	
System:Insert New Ticket • 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.	✓	
System:Insert Opportunity • Modified the script on the form.	✓	

Form Name / Change	7.5.1	7.5.2
System:Lead Detail • Modified the script on the form.	✓	
System:Manage Alternate Addresses • Modified the script on the form.	✓	
System:Manage Targets • 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.		✓
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.	✓	✓
System:Opportunity Detail • 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.		✓
System:Select Contact(s) • 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.	✓	
System:SLX Report Manager View • Modified the script on the form.	✓	✓
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 • Modified the Width of Label controls on the form. For the lveAcctManager:TLookupEdit control, set ReadOnlyEditor to True. Modified the script on the form.		✓
System:View History Attachments • For the grdAttach:TDataGrid control, modified the SQL property. Modified the script on the form.	✓	
Ticket:Attachments • For the grdAttach:TDataGrid control, modified the SQL property.	✓	
Ticket:Details • 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 "Finding Script Changes" on page 37 for more information on finding script changes.

Script Name	7.5.1	7.5.2
System:Global System	✓	

Updated VB Scripts

See "Finding Script Changes" on page 37 for more information on finding script changes.

Script Name	7.5.1	7.5.2
Dashboard:Content Set Support		✓
Dashboard:Export Support	✓	
Dashboard:Period Support		✓
System:Notes History Common	✓	
System:SLX Activity Support		✓
System:SLX Address Common		✓
System:SLX_Common		✓
System:SLX Crystal Report		✓
System:SLX Database Support		✓
System:SLX_Export_Group_To_Excel	✓	
System:SLX Lead Info	✓	
System:SLX Lead Mappings		✓
System:SLX Lead Support		✓
System:SLX Print Detail	✓	
System:SLX Report Conditions	✓	✓
System:SLX Report Condition Builder Controller	✓	
System:SLX Report Controller	✓	✓
System:SLX TimeZones		✓
System:SP_SalesProcessFunctions	✓	
Ticket:Ticket Activity Support		✓

Changes to Existing Groups

Name	7.5.1	7.5.2
CONTRACT:Active Contracts • (7.5.2) Modified for localization.	✓	✓
CONTRACT:All Contracts • (7.5.2) Modified for localization.	✓	✓

Name	7.5.1	7.5.2
CONTRACT:Pending Expirations • (7.5.2) Modified for localization.	✓	✓

Changes to Existing Menus and Toolbars

Name / Change	7.5.1	7.5.2
Menus System:Standard Menus • Added Create New Group item under the CampaignNavMenu > Groups item.	✓	

Changes to Existing Reports

Name / Change	7.5.1	7.5.2
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.	✓	