Applying Service Pack 4 for SalesLogix Version 6.2

Version 6.2.4

Developed by SalesLogix Technical Publications

Your business in mind.

Applying Service Pack 4 for SalesLogix Version 6.2.

Documentation Comments

This documentation was developed by SalesLogix Technical Publications. For content revisions, questions, or comments, contact the SalesLogix writers at techpubs@saleslogix.com.

Copyright

Version

Version 6.2.4 (031406)

Printed in the United States of America.

Trademarks

Disclaimer

Technical Support

Technical Support is available to customers with support contracts directly from Sage Software and to Certified Business Partners. Calls are answered during business hours, Monday through Friday, excluding holidays. Current contact information is available on the SalesLogix Web site. Customers with a valid technical support contract and a Web access code can request technical support electronically on the SalesLogix SupportOnline/Sage Online Support and Services Web site.

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

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

Contents

Introduction

This document provides instructions for applying Service Pack 4 for SalesLogix version 6.2. This service pack includes bug fixes, synchronization and conflict resolution changes, new licensing, and Sales Orders functionality that has been incorporated into the Advanced bundle.

Requirements

Before upgrading to version 6.2.4, you must:

- Ensure your current SalesLogix version is 6.2 or later.
- Create a database backup. Even if you back up every night, do another backup before this installation to preserve any last-minute changes. Then, verify that your backup completed successfully and that you can restore from it, if necessary.

About This Document

Chapter 2, “Changes in This Release” outlines the plugin changes from SalesLogix v6.2 to Service Pack 4. Use the Upgrade Workplan included in this document to guide your upgrade process.

If you have not upgraded to SalesLogix v6.2, refer to the following documents on the SupportOnline/Sage Online Support and Services Web site (http://www.sagesoftwareonline.com) to upgrade.

- Upgrading from SalesLogix Version 5.2.x to 6.2
- Upgrading from SalesLogix Version 6.x to 6.2

After upgrading, see the “What’s New in this Release” topic in the online Help systems for an explanation of the new features.

Chapter 1
Upgrade Workplan

Use this workplan to apply Service Pack 4 for SalesLogix v6.2. Follow the tasks in order. If a task does not apply, disregard it and move on to the next.

✓	Task	Notes
	Prepare to Upgrade	
	1 Read all chapters in this document before proceeding. Understanding the content of the release and planning your upgrade are critical.	

Task 1 Details

*** IMPORTANT PLEASE READ ***

Support for Oracle 8 has been discontinued

Do not install this service pack if you are currently running SalesLogix on Oracle 8. In version 6.2.4, support for Oracle 8 is discontinued. The functionality in this service pack will not run on Oracle version 8 or earlier.

Upgrading the Marketing Feature Pack

If your implementation currently includes the Marketing Feature Pack, you must distribute a new SalesLogix Client Help file to users after upgrading to v6.2.4. In addition, if you are running EmPulse with the Marketing Feature Pack, you must upgrade your EmPulse Server. If you do not upgrade your EmPulse Server, marketing components may not function correctly. The Feature Pack upgrade process is outlined in tasks 40-42.

Understanding the Service Pack .MSP files

The v6.2.3 release included two new files. Due to this change, multiple .MSP files are necessary when upgrading to v6.2.4. When upgrading from version 6.2, 6.2.1, or 6.2.2, the .MSP file adds the new files. When upgrading from version 6.2.3, the .MSP file updates the two new files.

The service pack installation automatically calls the correct .MSP file for your SalesLogix version. However, note that if you upgrade your Remote Clients using an .MSP file, you must select the correct file for your current version. See task 35 for details.

Understanding the Service Pack bundle extraction process

Depending on the installation method you choose while running the 6.2.4 service pack, bundles may or 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 zip file and you must browse to the bundles when you are ready to install them. However, 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 computer when you are ready to install them.

See task 19 to determine which option you want to select when installing the service pack and extracting bundles.

Running SLXSystem.exe as a Windows Service

In multi-user systems such as Citrix or Terminal Services, it is recommended that you run the SLXSystem.exe as a Windows Service. See tasks 6 and 29 for more information on this feature.

✓	Task	Notes
	2 Install SalesLogix v6.2.4 in a test environment before installing on your production database. **Note** Refer to SupportOnline/Sage Online Support and Services Knowledge Base article #03010106 for information on using a copy of your production database in a test environment.	
	3 (Optional) Record your customizations and use Bundler to create a bundle of all customized plugins in your database. When you create the custom bundle, ensure you select the Generate New ID on Install option for each of your plugins. **Note** SalesLogix does not overwrite your customized plugins with the v6.2.4 plugins. However, this bundle can be used as a backup of your customizations.	
	4 Understand the changes in v6.2.4. **Note** Service Pack 4 is a cumulative release and includes all changes released in v6.2.4, v6.2.3, v6.2.2, and v6.2.1. These changes are detailed in Tasks 4, 5, 6, and 7.	

Task 4 Details

In addition to defect fixes, v6.2.4 includes the following new functionality.

SQL Optimization Toolkit - Technology Preview

The SQL Optimization Toolkit adds the following features and functionality to SalesLogix:

- Provides a database independent mechanism for tuning SQL executed through the SalesLogix OLE DB Provider that does not require source level access to the SQL. You can use the Optimization Toolkit to hand-tune certain SQL statements. This may be used for troubleshooting performance issues that arise due to large databases and user sites.
- This Optimization Toolkit is integrated with the SalesLogix Profiling Toolkit. Individual queries can be selected and tuned, and the Provider substitutes the optimized query when any client application attempts to execute it.
- Optimized queries are database specific. Therefore, hand-tuned queries for each database vendor can co-exist in one database to support an Oracle host and Microsoft SQL remote environments.
- Optimizations can be turned on and off without having to remove them completely.

This is an unsupported technology preview.

✓	Task	Notes
	5 Understand the changes in v6.2.3. **Note** Service Pack 4 is a cumulative release and includes all changes released in v6.2.3.	

Task 5 Details

In addition to defect fixes, v6.2.3 included the following new functionality.

Virtual Server-Side Cursors

Virtual Server-Side Cursors were implemented in the SalesLogix Client to improve large group performance and memory usage in Groups and List views. The SalesLogix OLE DB Provider provides a server-side cursor implementation such that the SalesLogix Client can request data on an as-needed basis.

To troubleshoot group performance in the SalesLogix Client, an icon appears indicating if a Server-side or Client-side cursor is used for a particular group. For more information, see the "Virtual Server-Side Cursors" topic in the SalesLogix Client or Developers Reference Online Help.

Group Performance Updates

Version 6.2.3 improved large group performance and memory usage in the SalesLogix Client Groups and List views. This included an update to the Group Manager (for the Administrator only) to filter and hide groups for better performance and easier navigation.

Global ID Changes

By default, the GlobalID is set for each table used by the Logging Server. This change eliminated the need for the SalesLogix Client to manually set the GlobalID in script for new tables and simplified the processing needed in the Logging Server to determine the GlobalID programmatically. This functionality supports the Marketing Feature Pack and should not be edited.

New and Updated Architect Controls

Updated DataGrid Control - The following changes were made to the DataGrid control in Architect:

- Exposed the ShowSummaryFooter property for creating footers in the grid.
- Added the OnGetSummary event that occurs when calculating summaries.
- Added the OnFormatSummaryFooter event that fires when the DataGrid displays footer text.
- Exposed the RowAutoHeight property for text wrap in cells and allowing ToolTips on truncated cells.
- Exposed the OnCustomDrawCell event for cell level formatting.
- Added the ability to not specify a database column when used with unbound grids.

New Charting Control PREVIEW - Added new Charting control to the Architect for creating pie charts, bar charts, and cone charts. This control is a technology preview and will have full support in a future release.

New Windows Common Dialogs Control - Allows access to standard Windows dialog boxes in the SalesLogix Client.

SalesLogix Profiling Toolkit - Technology Preview

The SalesLogix Profiling Toolkit provides a database-independent profiling application to trace SQL and performance counters throughout the execution path of the OLE DB Provider. You can choose to profile any of the running applications that are using the SalesLogix OLE DB Provider. A key feature to this functionality is that you can profile any running application, rather than having to stop the application, set a registry setting, and restart the application.

This is an unsupported technology preview.

✓	Task	Notes
	6 Understand the changes in v6.2.2. **Note** Service Pack 4 is a cumulative release and includes all changes released in v6.2.2.	

Task 6 Details

Option to set the SLXSystem.exe to run as a Windows Service

In SalesLogix v6.2.2 and later, you can run the SLXSystem.exe as a Windows Service. This is highly recommended on multi-user systems such as Citrix, Web Server, or Terminal Services. Running SLXSystem.exe as a service removes the startup/shutdown time when the Client creates a connection to a database using the SLXSystem.exe.

After installing the service pack, you must shut down the current SLXSystem.exe in Task Manager and run a command line to install the application as a service. This process is detailed in task 29.

New Option for Creating Remote Databases with a SQL Server Host

In earlier versions of SalesLogix, remote databases could only be created using the Host SQL Server. In v6.2.2 and later, you can create remote databases using either the Host SQL Server or an instance of MSDE. Creating remotes on the Host is the quickest method, but using an instance of MSDE is the most secure. This new functionality matches the database creation functionality available for Oracle. See the “Create Database Options” topic in the Administrator Help for information on specific field changes.

Optional SalesLogix Client OLE DB Provider Installation

The SalesLogix Client OLE DB Provider installation allows you to install the client-side OLE DB Provider as a separate, standalone installation. This installation provides a solution for users who have third-party applications that need access to the SalesLogix Server, but do not want a SalesLogix Client installed on that machine. This installation is extracted with the service pack and is also available on SupportOnline/Sage Online Support and Services.

Optional SalesLogix Web Client Components Installation

The SalesLogix Web Client Components installation provides ActiveX capability for SalesLogix Web Client computers. The installation provides a solution for users who have disabled ActiveX control downloads in Microsoft Internet Explorer options, but who still want to use SalesLogix ActiveX components. This installation is extracted with the service pack and is also available on SupportOnline/Sage Online Support and Services.

✓	Task	Notes
	7 Understand the changes in v6.2.1. **Note** Service Pack 4 is a cumulative release and includes all changes released in v6.2.1.	

Task 7 Details

In addition to defect fixes, Service Pack 1 included synchronization and conflict resolution changes, new licensing, and the ability to retire users while upgrading.

New Concurrent User License

The Concurrent Support User license was renamed and modified to allow concurrent users to log on to the Web Client, SalesLogix Client, Support Client, Administrator, Architect, Sync Client (Remote Office only), and a Remote Office. Every logon consumes one Concurrent User license. For example, if Samantha Brink logs on to the Web Client and SalesLogix Client and leaves them both open, two licenses are consumed.

Concurrent User licenses can be divided between the Host and any or all Remote Offices. This assignment can be changed at any time (Administrator > Systems > Offices > Edit Remote Office > Sync Options tab). When you assign Concurrent User licenses to a Remote Office, the Host assumes all Remote Office licenses are consumed at all times. For example, if you have a 30 Concurrent User license and you assign 10 licenses to one Remote Office, the Host automatically subtracts the 10 licenses and only allows 20 Concurrent Users to log on to applications at the main office.

The upgrade to v6.2.4 converts Support Only licenses to Concurrent User licenses.

Web Host License Change

Web Host functionality (including Customer Portal), no longer requires a Premier license. An Advanced license is required for the Web Host.

Synchronization Changes

In v6.2.1 and later, when an error occurs during synchronization, a transaction is created containing the error and the information from the Transaction Exchange File (TEF). This transaction is written to a new folder named FailedTrans. For SalesLogix Remote Clients, the transactions are sent back to the Host and placed in the Host's FailedTrans folder. The TEF file can be opened with TrnViewer to troubleshoot the problem.

- **Remote SalesLogix Client**
  A folder named FailedTrans is created in C:\Documents and Settings\All Users\Application Data\SalesLogix\Sync. This folder is a temporary holding place for transactions that will be sent to the Host. At the end of a synchronization cycle, the Sync Client displays a message in the Event Log indicating one or more transactions failed to apply. For example:

  [10/20/2004 11:48:52 AM] 83 of 84 transactions from 14 file(s) were successfully applied to your local database. Your client data will now be refreshed. This will take a moment.
  [10/20/2004 11:48:52 AM] 1 transactions failed to apply!"

  After indicating there is a failed transaction, a message flashes stating that Sync Client is sending the failed transactions. Once the transactions are sent successfully, they are deleted from the FailedTrans folder on the remote.

  In addition to the new process, all errors continue to be logged to the SyncError.txt file.

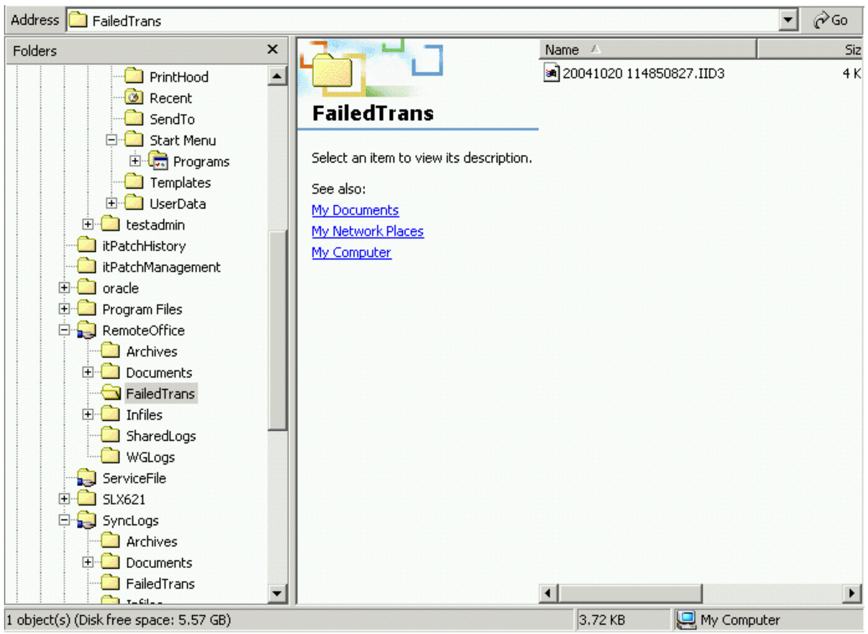
✓	Task	Notes

Task 7 Details - continued

- **Remote Office**

  The process is the same as that of a Remote SalesLogix Client. However, the FailedTrans folder is located in the Sync Log path for the Remote Office.

- **Host**
  - Failed transactions are logged to the FailedTrans folder in the Host synchronization path.
  - The FailedTrans folder at the Host may contain transactions from any Remote User, Remote Office, and the Host. Each transaction is formatted with the date/transaction ID/sitecode. For example, 20041020 114850827.IID3.
  - The system administrator can open the failed transaction files with TrnViewer.
  - Transactions in the FailedTrans folder at the Host must be deleted manually.



Conflict Resolution Changes

SalesLogix v6.2.1 changed the following process for conflict resolution:

- All Inserts except "Account Sends" have conflict resolution performed on them (to determine if the record already exists). Conflict resolution uses the values for the record already in the database to determine the winner. The losing record is removed from the database and the Insert is applied.
- When “Account Send” finds a record to apply that is already in the database, it removes the record and applies the new Insert.

Retiring Users While Upgrading

If you have more users than available licenses, you can retire users during an upgrade. Clicking the Retire Users button on the License Wizard automatically retires any excess users in your database. For example, if you have 35 Network Users and a 30 Network User license, 5 Network users will be retired during the upgrade.

When retiring users, SalesLogix follows all rules regarding Retired users (no managers, no account owners, etc.). SalesLogix will retire the first user it finds who is the correct user type. For example, if you have an excess number of Web Users, Web users will be retired.

✓	Task	Notes
	Task 7 Details - continued After applying licenses, the list of Retired users is available in the Administrator on the Retired tab of the Users view. View the retired users to determine if you need to: • Retire a different user and re-activate a user that SalesLogix has automatically retired. For detailed steps, see the “Changing User Type” (to retire a user) and “Activating or Deactivating a User Login” (to re-activate a user) topics in the Administrator Help. • Purchase additional licenses to activate Retired users. **Sales Orders Bundle Change** Sales Orders functionality has been incorporated into the Advanced bundle. Previously, this functionality was released in the Sales Orders bundle. **Web Regional Settings Change** In SalesLogix v6.2, the Web Client computers and Web servers were required to have the same regional settings. This requirement has been dropped in v6.2.1 and later. After applying the service pack, Web servers and Web Clients can use any mix of French, English, German, Italian, or Spanish regional settings without experiencing date problems.	
	8 Review Chapter 2, “Changes in This Release” to determine if you have customized any plugins that are included in the SalesLogix v6.2 Service Pack 4 bundle.	
	9 After reviewing the plugin changes, determine if you are going to add your customizations to the v6.2.4 plugins (recommended) or add the SalesLogix changes to your custom plugins. **Note** Some of the views and scripts in the bundles have had localization functions applied to them, but are otherwise unchanged. If you have customized these plugins, and you do not plan to install any of the SalesLogix localization bundles in the future, you can choose not to implement the localization changes.	
	10 If you currently have the Advanced bundle applied to your system, review “Changes to the Advanced Bundle” on page 34 to determine if you have customized any plugins that are included in the SalesLogix Advanced v6.2 Service Pack 4 bundle. Determine if you are going to add your customizations to the v6.2.4 plugins (recommended) or add the SalesLogix changes to your custom plugins. **Note** Some of the views and scripts in the bundles have had localization functions applied to them, but are otherwise unchanged. If you have customized these plugins, and you do not plan to install any of the SalesLogix localization bundles in the future, you can choose not to implement the localization changes.	
	11 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.	
	12 Run the Integrity Checker on your database. Integrity Checker is run from the Tools menu in SalesLogix Administrator. **Note** See the “Integrity Checker” topic in the Administrator Help for instructions.	

✓	Task	Notes
	Upgrade Your Main Office	
	13 Make sure all users have logged off SalesLogix.	
	14 Instruct all Remote Users and Offices to run a final synchronization cycle.	
	15 Execute a final cycle of all sync servers. The final sync cycle must run successfully and send out all pending transactions.	
	16 Stop the Sync Service(s). If using a third-party scheduling package, deactivate the Synchronization Server tasks.	
	17 Close all SalesLogix applications and stop all services that access the SalesLogix database using the SalesLogix OLE DB Provider.	
	18 Back up your production database. Verify the backup is successful and complete.	

✓	Task	Notes
	19 Run the SLX_v62_SP4 installation on your SalesLogix Server.	

Task 19 Details

You must run the service pack installation on every computer where a SalesLogix application is installed. This includes the SalesLogix Server, Administrative Workstation, SalesLogix Network Client, SalesLogix Remote Client, Support Client, Synchronization Server, SpeedSearch Server, Web Host, Web Reporting Server, Web Manager, and Remote Office.

To run the service pack installation

1. Double-click **SLX_v62_SP4.exe**.
2. On the **SalesLogix v6.2 Service Pack 4** screen, select your installation method:
   - **Install the SalesLogix v6.2 Service Pack 4** extracts the service pack files to a temporary folder and removes the installation files once the service pack installation is complete.
     If you select this option, the SalesLogix bundles will not be extracted to your local computer. You must browse out to the bundles in the zip file to apply them to your database.
   - **Extract and Install the SalesLogix v6.2 Service Pack 4** extracts the service pack files to a location you specify. The files are not removed once the installation is complete. If you select this option, choose a location where you want to store the service pack files. This option extracts the files and SalesLogix bundles required for the upgrade.

     If you have SalesLogix Remote Users, selecting the Extract and Install the SalesLogix Service Pack option allows you to distribute a smaller upgrade patch to your Remote Users. This process is detailed in task 35.
3. Click **Next**.
4. The **Welcome** screen displays a list of currently installed SalesLogix components. Verify the correct components appear in the list, then click **Install**.

   The Update Network Images button is available on the Welcome screen. If you created automated installations of the SalesLogix Client, you can upgrade the installations to service pack 4 using this option (you must browse to the .msi file and click Update).

   Automated installations can be used to install the SalesLogix Client for new users. When properly updated, this installation contains the initial version of the SalesLogix Client and all service packs and hot fixes that have been applied to your system.
5. When the installation indicates a successful upgrade, click **Finish**.

   The upgrade is successful when the Status column displays Complete.
6. It is recommended that you restart your computer after the installation to ensure that all changes take effect.

   You may automatically receive a prompt to restart.

✓	Task	Notes
	20 Run the SLX_v62_SP4 installation on your Administrative Workstation.	
	21 Start the SalesLogix Administrator.	

✓	Task	Notes
	22 Apply the Service Pack bundle named SalesLogix v6.2 Service Pack 4.sxb.	
	Task 22 Details If you extracted the service pack files to your local computer (task 19), the service pack bundle is located in the folder you specified. If you installed the service pack without extracting the files, you must browse out to the SLX_v62_SP4.zip file for the necessary bundle. The contents of the bundle are outlined in Chapter 2, "Changes in This Release". **To install the service pack bundle** 1 **SalesLogix Administrator** > **Bundles** > **Install**. 2 Browse to and double-click **SalesLogix v6.2 Service Pack 4.sxb**. 3 After the bundle is loaded, the **Choose Actions to Install** dialog box appears. View the plugins to be installed with this release, and then click **OK**. The Install column displays check marks indicating which plugins will be installed on your system. SalesLogix recommends installing all plugins. If you choose not to install some plugins, do not install any of the plugins listed under the relevant section of the bundle. For example, do not clear some plugins under ***Service Pack 1*** and leave others selected to be installed. During installation, click Yes or Yes to All on any confirmation message boxes for overwriting system plugins. Although this message indicates that your current plugins will be overwritten by the v6.2.4 plugins, they are not. SalesLogix versioning prevents plugins from being overwritten. 4 In the **Information** dialog box, the following message appears: "The following installed plugins already have releases. Do you want to update those releases?" • If you choose not to release upgraded plugins, your existing customizations remain released to users. • If you choose to release upgraded plugins, the v6.2.4 plugins will be released to users. Your customizations will not be overwritten, but they will be unavailable to users until you release them using Architect. 5 In the **Choose Teams** dialog box, select the teams to which you want to release plugins, and then click **OK**.	
	23 If your installation includes the Advanced bundle, apply the bundle named SalesLogix Advanced v6.2 Service Pack 4.sxb.	
	Task 23 Details While the bundle is installed, click Yes or Yes to All on any confirmation message boxes for overwriting system plugins. Although this message indicates that your current plugins will be overwritten by the v6.2.4 plugins, they are not. SalesLogix versioning prevents plugins from being overwritten. Disregard any messages regarding table indexes, and click OK.	
	24 Apply the v6.2.4 changes listed in Chapter 2, "Changes in This Release" You can use one of the following strategies: • Add your customizations to the v6.2.4 plugins (recommended). • Add the v6.2.4 changes to your custom plugins.	
	Task 24 Details When opening legacy plugins in Architect, you will receive a message indicating the plugin is read only. Make sure your company name is correct in Tools > Options, and then click Save to update the plugin to version 6.2.4.	

✓	Task	Notes
	25 After applying 6.2.4 changes, release the appropriate views and scripts using Architect. **Note** See the “Release a Plugin for Use” topic in the Architect Help for information on releasing plugins.	
	26 Run the SLX_v62_SP4 installation on your SpeedSearch Server.	
	27 Run the SLX_v62_SP4 installation on each of your SalesLogix Network user’s computers.	
	28 Run the SLX_v62_SP4 installation on each of your Support user’s computers.	
	29 If you have a multi-user system such as Citrix or Terminal Services, install the SLXSystem.exe as a Windows Service. **Note** For more information, see task 6.	
	Task 29 Details The SLXSystem service starts and stops when connections are created. If necessary, you can set the service to remain open for a set amount of time before shutting down. To set a specific amount of time for the service to remain open, add the registry key HKEY_LOCAL_MACHINE\SOFTWARE\SalesLogix\ShutdownDelay(DWORD). The key accepts values 1 through 1440, where the value indicates the number of minutes the service will run before shutting down (after it detects no activity). Setting the value to 0 indicates the service never shuts down. By default, the SLXSystem.exe service is set to Automatic with a registry value of 0 so that it is always on. **To install SLXSystem.exe as a service** 1 If currently running, close the current SLXSystem.exe using Task Manager. 2 On the **Start** menu, click **Run**. 3 In the **Open** box, type **“C:\Program Files\SalesLogix\SLXSystem.exe /install”**. The service can be removed using the “C:\Program Files\SalesLogix\SLXSystem.exe /uninstall” command.	
	Upgrade the Synchronization Server (if applicable)	
	30 Run the SLX_v62_SP4 installation on your Synchronization Server(s).	
	Task 30 Details Before installing the Service Pack, ensure the Synchronization Service (SLXSyncService) is stopped. In addition, shut down any instances of Synchronization Server in the Task Manager.	

<table>
<tr><th>✓</th><th>Task</th><th>Notes</th></tr>
<tr><td rowspan="2"></td><td>31 Create new folders and update the FTP synchronization directories in Administrator for the FailedTrans folder.
Note If you are upgrading from v6.2.1 or later, ignore this step.</td><td></td></tr>
<tr><td colspan="2">
Task 31 Details
FailedTrans information is detailed in task 7.
Creating the FailedTrans Folder
If you synchronize using FTP, you must create a folder and virtual directory for the FailedTrans folder.
1 Create a FailedTrans folder on your FTP server. For example, /FailedTrans. This folder must be shared, and all Remote Users and the Sync Server must have read, write, and delete access.
This folder should be in the same location as the /Infiles and /Outfiles folders.
2 Create a virtual directory that points to the /FailedTrans folder. All Remote Users and the Sync Server must have read, write, and delete access to this directory.
Updating FTP Directories in Administrator
After creating the FailedTrans folder and virtual directory, you must update the FTP Sync Transport Options in Administrator.
1 On the Administrator Manage menu, click Sync Transfer Profiles.
2 In the Sync Transfer Profiles dialog box, double-click your FTP sync profile.
3 In the Edit Sync Transfer Profile dialog box, click Setup.
4 Under Directories in the FailedTrans box, type the name of the FailedTrans directory. For example, /FailedTrans.
5 Click OK.
</td></tr>
<tr><td></td><td>32 Cycle your Sync 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.</td><td></td></tr>
<tr><td rowspan="2"></td><td>33 Update your Synchronization Service jobs.</td><td></td></tr>
<tr><td colspan="2">
Task 33 Details
To ensure compatibility with the SalesLogix Server, you must update your Synchronization Service jobs.
To update synchronization service jobs
1 In the Administrator, click Systems.
2 Click the Sync Automation Services tab.
3 Double-click the first job in the grid.
4 In the Database box, click the Open button.
5 In the Data Link Manager, select the connection to your SalesLogix database.
6 Click OK.
7 Click OK again.
8 Repeat until all jobs have been updated with the new database connection.
</td></tr>
<tr><td></td><td>34 Restart the Sync Service(s).</td><td></td></tr>
</table>

<table>
<tr><th>✓</th><th>Task</th><th>Notes</th></tr>
<tr><td></td><td>Upgrade SalesLogix Remote Users/Offices</td><td></td></tr>
<tr><td></td><td>35 Distribute the service pack installation to your Remote Users and Offices. You may want to distribute via e-mail or copy the installations to a shared network drive.</td><td></td></tr>
<tr><td></td><td colspan="2">Task 35 Details

To upgrade Remote Users and Offices you can distribute an .MSP file rather than the entire Service Pack patch. This file is smaller and upgrades only the necessary SalesLogix Client and Remote Office pieces. The SalesLogix .MSP files are located in the folder you specified when extracting the service pack files (see task 19).

After receiving the file, instruct the Remote User or Remote Office to double-click the appropriate .MSP file.

IMPORTANT

In Service Pack 4, you must run the .MSP file specific to your current SalesLogix version.
• If you are currently running SalesLogix v6.2, 6.2.1, or 6.2.2, use the SalesLogix Client 6.2.x to 6.2.4.msp file for Remote Users and the SalesLogix Remote Office 6.2.x to 6.2.4.msp file for Remote Offices.
• If you are currently running SalesLogix v6.2.3, use the SalesLogix Client 6.2.3 to 6.2.4.msp for Remote Users and the SalesLogix Remote Office 6.2.3 to 6.2.4.msp file for Remote Offices.</td></tr>
<tr><td></td><td>36 Direct Remote Users and Offices to run the SLX_v62_SP4 installation (or the appropriate .MSP file) on their remote computers.</td><td></td></tr>
<tr><td></td><td>37 Direct Remote Users to start the SalesLogix Client.</td><td></td></tr>
<tr><td></td><td>38 Instruct Remote Users and Offices to sync immediately after upgrading.</td><td></td></tr>
<tr><td></td><td>Upgrade Web Components (if applicable)</td><td></td></tr>
<tr><td></td><td>39 If your SalesLogix implementation includes Web Components, refer to the Upgrading the SalesLogix Web Components to v624.pdf for upgrade instructions.</td><td></td></tr>
</table>

<table>
<tr><th>✓</th><th>Task</th><th>Notes</th></tr>
<tr><td></td><td>Upgrade Marketing Feature Pack Components (if applicable)</td><td></td></tr>
<tr><td></td><td>40 Understand the Marketing Feature Pack changes.</td><td></td></tr>
<tr><td></td><td colspan="2">Task 40 Details

The SalesLogix Marketing Feature Pack is provided with the v6.2.4 Service pack for both current v6.2.3 installations that contain Marketing and v6.2.4 installations that want to install the Marketing Feature Pack.

The Marketing Feature Pack update contains provider fixes that are necessary when running Marketing with EmPulse on SalesLogix v6.2.4. If you are running Marketing with EmPulse and do not update your EmPulse Server, EmPulse components may not function correctly. If you are running Marketing without the EmPulse components, you do not need to run the EmPulse installation.

The v6.2.4 Marketing Feature Pack contains the following components:
• Marketing Feature Pack bundle. Do not apply this bundle if you are upgrading your Marketing implementation. There are no plugin or schema differences in this bundle between v6.2.3 and v6.2.4. The bundle was updated only to remove the SalesClient.chm Help file. The bundle is provided for new installations of the Marketing Feature Pack only.
Caution: If you reapply the Marketing Feature Pack bundle when upgrading, you may introduce problems with your Campaign data.
• Web Marketing Feature Pack bundle. This bundle is unchanged from v6.2.3. The bundle is provided for new installations of the Marketing Feature Pack only.
• SalesLogix EmPulse v6.2.4.exe. This patch must be run to update your EmPulse Server. See task 41 for details.
• SalesLogix Web Marketing Feature Pack v6.2.3.exe. This patch is unchanged from v6.2.3. The installation is provided for new installations of the Marketing Feature Pack only.
• SalesLogix Client Help file and documentation. The documentation was updated to reflect v6.2.4 changes.</td></tr>
</table>

✓	Task	Notes
	41 If necessary, upgrade your EmPulse Server. • If your installation includes the Marketing Feature Pack and you are using EmPulse components, you must upgrade your server. • If your installation includes the Marketing Feature Pack and you are not using EmPulse components, you do not need to run the upgrade. • If your installation does not include the Marketing Feature Pack, you do not need to run the upgrade.	

Task 41 Details

Run SalesLogix EmPulse v6.2.4.exe on your EmPulse Server. You may have installed EmPulse on your SalesLogix Server.

To run the EmPulse installation

1. Double-click **SalesLogix EmPulse v6.2.4.exe**.
   By default, the installation is extracted to the Marketing Feature Pack v6.2.4 folder within the temporary folder you specified when extracting the service pack.
2. On the **SalesLogix EmPulse v6.2.4** screen, select your installation method:
   - **Install SalesLogix EmPulse v6.2.4** extracts the Marketing Services files to a temporary folder and removes the installation files once the installation is complete.
   - **Extract and Install SalesLogix EmPulse v6.2.4** extracts the Marketing Services files to a location you specify. The files are not removed once the installation is complete. If you select this option, choose a location where you want to store the files.
3. Click **Next**.
4. On the **Welcome** screen, click **Next**.
5. On the **Program Maintenance** screen, select **Modify**, and then click **Next**.
6. On the **Custom Setup** screen, click **Next**.
7. On the **Ready to Modify the Program** screen, click **Install**.
8. When the installation is complete, click **Finish**.

✓	Task	Notes
	42 Update the SalesLogix Client Help. **Note** If your installation includes the Marketing Feature Pack (with or without EmPulse components), you must distribute an updated SalesClient.chm Help file to all users. **Task 42 Details** If your implementation does not include the Marketing Feature Pack, do not distribute the Help file. The 6.2.4 Service Pack automatically installs the SalesLogix Client Help file that does not include Feature Pack information. If you have the Marketing Feature Pack installed, the 6.2.4 Service Pack overwrites the Help file containing Feature Pack information with legacy Marketing Help. Therefore, you must reapply the SalesLogix Client Help file that contains Marketing Feature Pack information. **To distribute the Marketing Help file** 1 Browse to and locate the **SalesClient.chm** file. By default, the Help file is extracted to the Marketing Feature Pack v6.2.4 folder within the temporary folder you specified when extracting the service pack. 2 Distribute the Help file to all SalesLogix Client users. • For Network users, you may want to send the file via e-mail and instruct users to save the file to the ...\Program Files\SalesLogix folder on their computer. Optionally, the file can be copied to a shared network folder from which users can copy and paste the file to their computer. You might also consider creating a self-extracting zip file. • For Remote users, you may want to create a Remote Task to send the Help file to all remotes. See the “Adding a Remote Task” topic in the Administrator Help for details. 3 When prompted, users should overwrite the existing Help file with the new file.	
	Perform Post-Upgrade Tasks	
	43 If necessary, update your SalesLogix installation with Concurrent User licenses. **Note** If you are upgrading from v6.2.1 or later, ignore this step. **Task 43 Details** If you plan to add Concurrent Users to your SalesLogix implementation, you may need to perform additional administrative tasks. If necessary, refer to the Help topics referenced in the following scenarios for more information. • If you are converting existing users to Concurrent Users, see the “Changing User Type” topic in the Administrator Help. • If you have Concurrent licenses and a Remote Office and you plan to assign Concurrent users to the Remote Office, see the “Office Sync Options Tab” topic in the Administrator Help.	
	44 Instruct users to log on to SalesLogix.	
	45 If necessary, in the location where you extracted the service pack (workplan task 19), delete the extracted service pack files. These files are automatically deleted if you selected the Install the Service Pack option.	

Chapter 2
Changes in This Release

In addition to the new functionality described in Workplan tasks 4, 5, 6, and 7, this service pack fixes product defects. This includes defects previously addressed in hot fixes.

Refer to the Fixed Issues List for v6.2.4 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.

Schema Changes

A check mark indicates the version number in which the schema change was initially released.

Schema Change	6.2.1	6.2.2	6.2.3	6.2.4
Create Index CONTACT_EMAIL3 on CONTACT (EMAIL3) • Improves the time needed to use the SendSLX e-mail option.		✓		
Delete From PICKLIST Where ITEMID In ('kQF8AA0000M2', 'kQF8AA0000M3', 'kQF8AA0000M4')	✓			
DELETE FROM PICKLIST WHERE TEXT = 'Opporunity'	✓			
Delete from PLUGIN where FAMILY = 'Ticket' And NAME = 'Support Ticket History' And TYPE = 24 • Removes the legacy plugin named Ticket:Support Ticket History.	✓			
drop index ticketactivity.XPKTICKETACTIVITY	✓			
drop index USERINFO.XPKUSERINFO	✓			
INSERT INTO SLXTRIGGERS(TABLE_NAME, TYPE, SEQUENCE, PATH, FUNCTION_NAME, ENABLED, VERSION, COLUMN_IDS) VALUES('SLXQUERYPLANHINT', 3, 1, 'SLXTriggers.dll', 'ExecuteOptimizerHintRefresh', 'T', 1, '*'				✓
INSERT INTO SLXTRIGGERS(TABLE_NAME, TYPE, SEQUENCE, PATH, FUNCTION_NAME, ENABLED, VERSION, COLUMN_IDS) VALUES('SYSTEMINFO', 1, 1, 'SLXTriggers.dll', 'ExecuteOptimizerMode', 'T', 1, 'OPTIMIZERMODE'				✓
INSERT INTO sysdba.SECCODE (SECCODEID, SECCODEDESC, SECCODETYPE) VALUES ('DEPT00000000', 'None', 'D') • Inserts the "None" department into SECCODE that was added in v6.2 base databases.		✓		
update branchoptions set licnum = '54C891A2B789CA'	✓			
update joindata set secondary = 'T' where totable = 'picklist'			✓	
UPDATE RESYNCTABLEDEFS SET OMNIDIRECTIONAL = 'T' WHERE TABLENAME = 'QUOTA'		✓		

Schema Change	6.2.1	6.2.2	6.2.3	6.2.4
update Resynctabledefs set omnidirectional = 'X' where tablename = 'SyncSequencing'		✓		
update Resynctabledefs set secondaryfield = 'TARGETSITE' where tablename = 'SyncSequencing' and secondaryfield is null		✓		
UPDATE SECTABLEDEFS SET AUTOINCREMENT = 'T' WHERE TABLENAME = 'SLXQUERYPLAN' AND FIELDNAME = 'SLXQUERYPLANID'				✓
UPDATE SECTABLEDEFS SET AUTOINCREMENT = 'T' WHERE TABLENAME = 'SLXQUERYPLANHINT' AND FIELDNAME = 'SLXQUERYPLANHINTID'				✓
UPDATE SECTABLEDEFS SET AUTOINCREMENT = 'T' WHERE TABLENAME = 'SLXTRIGGERS' AND FIELDNAME ' 'ID'				✓
UPDATE SECTABLEDEFS SET DATETIMETYPE = 'D' WHERE TABLENAME = 'CAMPAIGN' AND FIELDNAME = 'ACTUALLAUNCHDATE' • Added to correctly display the date value in the Web Client.	✓			
UPDATE SECTABLEDEFS SET DATETIMETYPE = 'D' WHERE TABLENAME = 'CONTACTRESPONSE' AND FIELDNAME = 'RESPONSEDATE'	✓			
UPDATE SYSTEMINFO SET DBVERSION = '6.21' WHERE SYSTEMINFOID = 'PRIMARY'	✓			
UPDATE SYSTEMINFO SET DBVERSION = '6.22' WHERE SYSTEMINFOID = 'PRIMARY'		✓		
UPDATE SYSTEMINFO SET DBVERSION = '6.23' WHERE SYSTEMINFOID = 'PRIMARY'			✓	
UPDATE SYSTEMINFO SET DBVERSION = '6.24' WHERE SYSTEMINFOID = 'PRIMARY'				✓
UPDATE SYSTEMINFO SET OPTIMIZERMODE='F'				✓
UPDATE USERSECURITY SET USERCODE = 'AllAccess' WHERE USERCODE = 'All Access'	✓			
UPDATE USERSECURITY SET USERCODE = 'CustomerService' WHERE USERCODE = 'Customer Service'	✓			
UPDATE USERSECURITY SET USERCODE = 'CustomerServiceManager' WHERE USERCODE = 'Customer Service Manager'	✓			
UPDATE USERSECURITY SET USERCODE = 'CustomerSupport' WHERE USERCODE = 'Customer Support'	✓			
UPDATE USERSECURITY SET USERCODE = 'CustomerSupportManager' WHERE USERCODE = 'Customer Support Manager'	✓			
UPDATE USERSECURITY SET USERCODE = 'ExecutiveManagement' WHERE USERCODE = 'Executive Management'	✓			
UPDATE USERSECURITY SET USERCODE = 'SalesManager' WHERE USERCODE = 'Sales Manager'	✓			

The following tables are created when installing the service pack bundle.

New Table	6.2.1	6.2.2	6.2.3	6.2.4
SLXQUERYPLAN • Necessary for the SQL Optimization Toolkit.				✓

New Table	6.2.1	6.2.2	6.2.3	6.2.4
SLXQUERYPLANHINT • Necessary for the SQL Optimization Toolkit.				✓
SYNCSECTABLES			✓	

The following fields are created when installing the service pack bundle.

Table	New Field	6.2.1	6.2.2	6.2.3	6.2.4
ACCOUNTPRODUCT	OPPORTUNITYID, OPPPRODUCTID, QUANTITY		✓		
BRANCHOPTIONS	LICNUM	✓			
RESYNCTABLEDEFS	DEFAULTGLOBALIDPATH • Reserved for the v6.2.3 Feature Pack for Marketing.			✓	
SECTABLEDEFS	AUTOINCREMENT	✓			
SYSTEMINFO	OPTIMIZERMODE • Added for the Optimizer preview. Value is set to F.				✓
USERINFO	ACCOUNTINGUSERID	✓			

New Forms

A check mark indicates the version number in which the form was initially released.

Form Name	6.2.1	6.2.2	6.2.3	6.2.4
System:Add Edit Address	✓			
System:Do Not Prompt Msg Box			✓	
System:Manage Alternate Addresses • The Manage Alternate Addresses form was converted from Legacy to ActiveX.	✓			

New Scripts

A check mark indicates the version number in which the script was initially released.

Script Name	6.2.1	6.2.2	6.2.3	6.2.4
Opportunity:SLX OnOpen Opportunity • Added in v6.2.3.4.				✓
System:Global System			✓	
System:SLX Address Common	✓			
System:SLX Contact Account Support • Created as a replacement for the existing legacy API's AddContactAndAccount, AddContactForAccount, and AddPersonalContact. When used as an included script, you can directly launch the active form for adding new contacts and accounts. This bypasses the dialog box where a user chooses if he/she is adding a new contact for an account, a new contact and account, a new account, or a new personal contact.			✓	
System:SLX_Multi_Currency	✓			

Script Name	6.2.1	6.2.2	6.2.3	6.2.4
System:SLX_Opportunity_Associate_Product_Set_Description	✓			
System:Opportunity Management	✓			

New Joins

A check mark indicates the version number in which the join was initially released.

Join / Change	6.2.1	6.2.2	6.2.3	6.2.4
PACKAGEPRODUCT/PRODUCTID to PRODUCT/PRODUCTID			✓	

Changes to Existing Main Views

A check mark indicates the version number in which the change was initially released.

Main View Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
System:Account Details • (6.2.1) Included the System:SLX Address Common script on the form. For the GroupsPane:TGroupsPane control > PopupMenu property > &Write E-Mail to Selected Records... menu item, set Argument to MainView:AdWriteEmailtoSelectedRecords. • (6.2.2) In the script on the form, added the OnFunctionEnabled_ViewAltAddress function to disable the View Alternate Address menu item when the view is read-only.	✓	✓		
System:Contact Details • (6.2.1) For the GroupsPane:TGroupsPane control > PopupMenu property > &Write E-Mail to Selected Records... menu item, set Argument to MainView:AdWriteEmailtoSelectedRecords. • (6.2.2) In the script on the form, added the OnFunctionEnabled_ViewAltAddress function to disable the View Alternate Address menu item when view is read-only.	✓	✓		
System:Opportunity Details • No changes.	✓			
System:Ticket Details • In the script on the form, modified the GroupsPanePopupMenuClick(Sender, MenuItem) routine to change Write:Email to MainView:AdWriteEmailtoSelectedRecords.	✓			

Changes to Existing Forms

A check mark indicates the version number in which the change was initially released.

Form Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
Account:Contracts • (6.2.1) For the frmContractAccounts:TAXForm control, set Width to 451. For the splGrid:TSplitterPanel control, set Width to 439. • (6.2.4) In the script on the form, modified for localization.	✓			✓
Account:Notes-History • For the cmdOptionsFull:TButton control, set Top to 2.	✓			

Form Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
Account:Opportunities • (6.2.1) In the script on the form, included the System:SLX Database Support script. Added localized calls to the UpdateStatusBarTotals function and wrapped the SLXIsNumeric function call around the result. In the grdAccountOppsPopupMenuClick(Sender,Item) sub, added code to check for version 6.2 Insert Opportunity view or else launch the legacy opportunity wizard. In the CallAccountDetailsView sub, added a MaxCount value. In the grdAccountOppsFormatColumnText function, added CreateDate field to the grid query. CreateDate is needed if the DateOpened value is NULL (which it may be if the Opportunity is coming from a pre- v6.2 system). The grid will use CreateDate if the DateOpened is NULL. • (6.2.2) In the script on the form, added If Not grdAccountOpps.Recordset.EOF statement to the UpdateStatusBarTotals function. Added If (IsDate statements to the grdAccountOppsFormatColumnText function. • (6.2.3) No changes.	✓	✓	✓	
Account:Reseller Opportunities • (6.2.1) Modified for localization. • (6.2.2) For the grdOpportunities:TDataGrid control, set the OnFormatColumnText event to grdOpportunitiesFormatColumnText. In the script on the form, included the System:SLX Common script. In the AXFormChange(Sender) sub, modified the code so calculations can be done correctly for Days Open. Added the grdOpportunitiesFormatColumnText function. Modified for localization.	✓	✓		
Account:Tickets • The wrong calculated field was being used (FullName). Changed to LastFirst.	✓			
Contact:Notes-History • For the cmdSendToWord:TButton control, set the Picture property to the Word icon that is used on other Notes-History views.	✓			
Contact:Opportunities • In the script on the form, modified the OpenOpportunityRolesView sub to call GetContactDisplayNameFromID which returns the name formatted Last First. In the CallOpportunityDetailView sub, added a MaxCount value. In the grdOppContactsFormatColumnText function, added check for Null values.	✓			
Contact:Tickets • In the script on the form, modified the CallInsertNewTicketView sub to reference the MVInsertNew.DetailsView.	✓			
Opportunity:Competitors • (6.2.2) In the script on the form, modified for localization. • (6.2.3) In the script on the form, modified the txtDataBoundChange sub so that setting the RowSelect property will make the grid Read Only and disable the DisableEditor property for all columns. Modified the grdCompetitorsDblClick sub to include Not grdCompetitors.RowSelect.		✓	✓	
Opportunity:Contacts • (6.2.1) In the script on the form, modified the CallContactDetailView sub to set a maximum count of 250. • (6.2.3) In the script on the form, modified the AXFormChange sub so that setting the RowSelect property will make the grid Read Only and disable the DisableEditor property for all columns. • (6.2.4) In the script on the form, added 'Application.BasicFunctions.SetCurrentContactID(grdOppContact.GetCurrentField("CONTACTID")) to grdOppContactPopupMenuClick sub. In the EnableDisableGridButtons sub, set grdOppContact.PopupMenu.Items(1)Enabled = blnEnabled.	✓		✓	✓

Form Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
Opportunity:Notes-History • (6.2.1) For the memNotes:TMemo object, set Width to 2. • (6.2.2) Modified for localization.	✓	✓		
Opportunity:Order Details • Modified for localization.		✓		
Opportunity:Products • (6.2.1) In the script on the form, modified the CallAddOpportunityProductsView sub for multi-currency. Modified for localization. • (6.2.2) In the script on the form, modified the CommitChanges sub for the Round function. Modified the CallAddOpportunityProductsView sub for multi-currency. • (6.2.3) In the script on the form, modified the txtDataBoundChange sub so that setting the RowSelect property will make the grid Read Only and disable the DisableEditor property for all columns. Modified the EnableDisableGridButtons sub for menu items.	✓	✓	✓	
System:Account Detail • (6.2.1) For the DummyLookup:TLookupEdit object, set Left to 317 and Top to 97. For the lblCurrentAddress:TLabel object, set Height to 44 and Width to 52. For the memBusDes:TMemo object, set Left to 517. For the mnuAddress:TPopupMenu object, set Left to 67. For the txtAcountID:TEdit object, set Left to 300 and Top to 97. Added the txtAddressID:TEdit object, set Height to 10, Left to 334, Name to txtAddressID, TabOrder to 41, Text to Account:AddressID, Top to 97, Visible to False, and Width to 10. In the script on the form, included the SLX Address Common script. • (6.2.2) In the script on the form, modified the CallManageAlternateAddressesView function to determine if the account is Read Only for the Current User. Modified the mmoAddressDblClick sub. Modified for localization. • (6.2.4) In the script on the form, modified the cmdAddressClick(Sender) sub (removed gStrCrntAddressID and replaced with lblCurrentAddress.Caption). Modified for localization.	✓	✓		✓
System:AddAddressDialog • In the script on the form, modified the UpdateAddress sub to check the origin of the request before prompting.	✓			
System:Add Center • Modified for localization (changed EditBoxes to Label controls and replaced Panels with Bevels).		✓		
System:Add Edit Competitor • (6.2.1) In the script on the form, modified the GetSelectedCompetitor sub for the txtCompetitor.Text line. For the memWeakness:TMemo object, set MaxLength to 128. • (6.2.2) Modified for localization.	✓	✓		
System:Add Edit Product Package • For the grdMembers control, set the OnEdited event to grdMembersEdited. In the script on the form, modified the cmdButtonClick sub and the DefinePackagesGrid function.	✓			

Form Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
System:Add Edit Ticket Activity • (6.2.1) Modified the tab order of controls on the form. For the cmdComments:TButton object, set Alignment to taCenter and Flat to disabled. For the memComments:TMemo control, removed the OnClick event, set Left to 73, and ScrollBars to ssVertical. In the script on the form, removed the Ticket:Ticket Activity Support and System:Spell Check scripts. In the Init(strMode) function, added dteDateStart = dteReceivedDate.DateTime. Modified the cmdOKClick(Sender) and CompleteDateChange subs. Removed code from the AddDateTimeStamp sub. Modified the memCommentsMouseDown sub to disable the Spell Check menu if Word is not installed. • (6.2.2) In the script on the form, in the Init(strMode) function, added gStrChargeType = leChargeType.Text. Modified the cmdOKClick(Sender), leChargeTypeChange(Sender), and CompleteDateChange subs.	✓	✓		
System:Add Edit Ticket Rate • Modified the tab order of controls on the form. In the script on the form, modified the txtAmountExitControl(Sender) sub to include Sender.Text <> "" And.	✓			
System:Add New Contact Account • (6.2.1) For frmAddNewConAcc:TAXForm, removed the OnDestroy event. For the lueLeadSource:TLookupEdit control, removed dddd from the FormatString property, set HideButtonIfReadOnly to True, set Lookup to LEADSOURCE:Description, and LookupID to blank. For the txtFax:TEdit control, set the OnExitControl event to txtFaxExitControl. In the script on the form, included the System:SLX Address Common script. Modified the creation order within the script so that an address is created first, which returns the primary AddressID and/or ShippingID. Then the account/contact record is created. Set focus on the account address memo box when returning from a modal dialog box. Removed all calls to LogSetGlobalID and LogClearGlobalID. Changed the procedure to call GetDisplayAddress function and removed the repetitive calls. Updated the Description, Salutation, and Zip field values in the multiple sub routines. • (6.2.2) Modified for localization. • (6.2.4) Modified for localization.	✓	✓		✓
System:Add Opportunity Product • (6.2.1) For the cmdOK:TButton control, set Default to false and Kind to bkCustom. For the ppeExchangeRate:TPopupEdit control, set the OnExitControl event to ppeExchangeRateExitControl and the OnKeyPress event to ppeExchangeRateKeyPress. In the script on the form, modified the DefineManagedPrdRS sub for KeyFieldID. Modified the AddItemsToTree(strSQL) sub to disable the Manage Products button if the user does not have permissions to manage products. Modified the GetSelectedPackageInfo sub to ReplaceSingleQuote prior to executing the query. Removed code from the AddProductToGrid sub. Modified the RemoveProductsFromGrid sub to remove .Item. In the ppeExchange RatePopup sub, added an if statement to check for ppeExchangeRate.text being equal to " " and setting it to “1” if blank. • (6.2.2) In the script on the form, in the cmdButtonClick sub, changed end to End. Modified the grdProductsPopupMenuClick sub to first select all records in the grid, then call the RemoveProductsFromGrid procedure which will update Managed records and reflect the change in the grid. • (6.2.3) No changes.	✓	✓	✓	

Form Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
System:ChooseAddContactAccount • (6.2.2) In the script on the form, in the AXFormCreate(Sender) sub, added code for enabling/disabling buttons depending on which radio button is selected. • (6.2.3) In the script on the form, in the cmdNextClick(Sender) sub, set Application.MainViews.AddEx "System:Contact Details" and Application.MainViews.AddEx "System:Account Details" set open existing to False.		✓	✓	
System:Close Opportunity • (6.2.1) In the script on the form, in the AddAccountProductRecords sub, modified so that if the opportunity has a sales order associated to it, do not create AccountProduct records. The sales order creates the records when the sales order is closed. Modified for localization. • (6.2.3) In the script on the form, modified the txtActualAmtExitControl(Sender) sub to check for a Null value. If the Actual Amount field is blanked out, place a "0" in the field. • (6.2.4) Modified for localization.	✓		✓	✓
System:Contact Detail • (6.2.1) For frmContactDetail:TAXForm, set the OnClose event to AXFormClose. For the mnuAddress:TPopupMenu control, set Left to 514 and Top to 44. Added the txtAddressID:TEdit control. In the script on the form, included the System:SLX Address Common script. Removed the SetAddress(strAddressID) sub. Modified the AXFormChange(Sender), FunctionLookupClick(Sender), ButtonShow(Sender), ButtonHide(Sender), and cmdMapquestClick(Sender) sub. Added mnuAddressClick sub. • (6.2.2) In the script on the form, modified the cmdAddressClick(Sender) sub to use the CurrentAddressID. Modified the CallManageAlternateAddressesView function to determine if the account is Read Only for the current user.	✓	✓		
System:Contract Detail • (6.2.1) For the grdAttach:TDataGrid control, removed the sort on the Attachment column and added a descending sort on the Last Modified Date column. In the script on the form, modified the cmdSaveCopyClick(Sender) sub. Added the CalculateContractListPrice(Sender) sub. Modified the pklContractTypePopupReturn(Sender) sub. Modified the memCommentsMouseDown sub to enable the Spell Check menu item if Word is installed. Modified for localization. • (6.2.2) In the script on the form, modified the Init(ContractID) function to check for an empty string that was handled in the CheckForExpiredContract procedure. Modified for localization.	✓	✓		
System:Copy Quota • No changes.		✓		
System:Edit Opportunity Competitor • For the memNotes:TMemo control, set MaxLength to 8000. For the memStrategy:TMemo, memStrengths:TMemo, and memWeakness:TMemo controls, set MaxLength to 128.	✓			
System:Edit Opportunity Contact • For the mrvSchedules:TPopupMemo control, in the Items property, removed the E-mail menu item. In the script on the form, removed the e-mail menu item logic from the mrvSchedulesClick(Sender, ByRef MenuItem) sub.	✓			
System:Edit View Notes History • (6.2.1) For the memNotes:TMemo control, set ScrollBars to ssVertical and Width to 440. For the cmdCancel:TButton, cmdHelp:TButton, and cmdOK:TButton controls, set Left to 13. For the pnlButtons:TPanel control, set Left to 440 and Width to 98. • (6.2.2) For Form:TAXForm, set HelpContext to 1678.	✓	✓		

Form Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
System:Insert New Ticket • (6.2.1) For the leArea:TLinkEdit and leCategory:TLinkEdit controls, set the OnEnterControl event to OnACIEnterControl, set the OnExitControl event to OnACIExitControl, and the OnPopup event to OnACIPopup. For the leIssue:TLinkEdit control, set the OnEnterControl event to OnACIEnterControl and the OnPopup event to OnACIPopup. For the lveAssignedTo:TLookupEdit control, set LookupRestrictField to SECCODETYPE, LookupRestrictOp to <>, and LookupRestrictValue to D. In the script on the form, included the System:ACI Support script. Added Dim strACIMatchText and Dim strCrntACIValue values. Added code to the ValidateForm function to populate Account if Account is empty when selecting a Contact. Removed the CallSLXACISelectView(Sender) sub. Added the OnACIEnterControl(Sender), OnACIExitControl(Sender) and OnACIPopup(Sender) subs. Modified the OnACIChange(Sender) sub. Modified the OnMenuMouseDown(Sender, Button, X, Y) sub to disable the Spell Check menu item if Word is not installed. • (6.2.2) In the script on the form, removed the CheckContractType sub. Modified the SetDefaults sub to include strItemID and strItem. Modified for localization. • (6.2.3) For the lveAssignedTo: TLookupEdit control, set LookupMode to lmOwner. Modified for localization. • (6.2.4) Modified for localization.	✓	✓	✓	✓
System:Insert New Ticket - Account Service Information • For the lveAccount:TLookupEdit control, set LookupID to Account:AccountID. In the script on the form, modified the grdContractFormatColumnText(Sender, ByRef Column, FormattedText, RawText, RowValuesArray) function.	✓			
System:Insert Opportunity • (6.2.1) For the cmdOK:TButton control, removed the OnClick event. For frmInsertOpportunity:TAXForm, set the OnCloseQuery event to AXFormCloseQuery. For the lveAccount:TLookupEdit control, removed the OnChange event. For the pklCloseProb:TPickList control, set ReadOnly to False. For the ppeExchangeRate:TPopupEdit control, set OnExitControl to ppeExchangeRateExitControl and OnKeyPress to ppeExchangeRateKeyPress. In the script on the form, modified the AXFormOpen(Sender), lveAccountPopupReturn(Sender), CallAddOpportunityProductsView, SaveOpportunityDetailInfo, DefineCompetitorsGrid, LoadContactGrid(objRS), ppeExchangeRatePopup(Sender), and AddSelectedCompetitors(objDetail) subs. Removed the lveAccountChange(Sender) sub. Added the ppeExchangeRateExitControl(Sender) sub. Modified for localization. • (6.2.2) In the script on the form, added the RefreshTabMiddleViews function to the SaveOpportunity(Sender) sub. Modified the lveAccountPopupReturn(Sender), LoadDefaultInfo, SaveOpportunityDetailInfo, SaveOpportunityProductInfo, SaveOpportunityCompetitorInfo, LoadContactGrid(objRS), and ppeResellerPopup(Sender) sub. • (6.2.3) No changes. • (6.2.4) In the script on the form, in the lveAccountPopupReturn(Sender) sub, removed the logic to use the Account ID based on the Account name. Modified the DefineContactsGrid sub to change the WORKPHONE field in the Opportunity Contacts grid from 24 to 32 characters. Modified for localization.	✓	✓	✓	✓
System:Manage Area Category Issue • In the script on the form, modified the EnableDisableButtons, cmdDeleteClick(Sender), and cbIssueTextClick(Sender) sub.	✓			

Form Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
System:Manage Customer Service Defaults • (6.2.1) In the script on the form, modified the DefineUserDefaultsGrid and PopulateUserDefaultsGrid(objRS) sub. • (6.2.2) Modified for localization.	✓	✓		
System:Manage Customer Service User Defaults - Individual • For the lblTicketActivity:TLabel control, in the Caption property, corrected spelling of the word Calendar.	✓			
System:Manage Customer Service User Defaults - Multiple • For the chkTicketActivity:TCheckBox control, in the Caption property, corrected spelling of the word calendar.	✓			
System:Manage Product • (6.2.1) In the script on the form, modified the Dim aButtonArray to 6. Modified the RefreshViews, BuildArrayOfGridButtons, and EnableDisableButtons subs. • (6.2.3) In the script on the form, modified the DeleteSelectedProduct sub to call CascadeDelete.	✓		✓	
System:Manage Product Packages • (6.2.1) In the script on the form, modified the CallAddEditProductPackageView(Mode) sub. • (6.2.2) Modified for localization.	✓	✓		
System:Manage Quota • (6.2.1) In the script on the form, modified the EnableDisableButtons sub. • (6.2.2) In the script on the form, modified the PopulateGrids, grdSalesClick(Sender), AddNewQuotaRecord, DeleteQuotaRecord, and EnableDisableButtons subs.	✓	✓		
System:Memo PopUp • In the script on the form, modified the memTextMouseDown(Sender, Button, X, Y) sub.	✓			
System:Opportunity Defaults • (6.2.1) In the script on the form, modified the cmdProductsClick(Sender) sub. • (6.2.2) No changes. • (6.2.3) No changes.	✓	✓	✓	
System:Opportunity Detail • (6.2.1) For the pklProbability:TPickList control, set the OnEnterControl event to pklProbabilityEnterControl. For the pklStatus:TPickList control, set the OnEnterControl event to pklStatusEnterControl and the OnExitControl event to pklStatusExitControl. In the script on the form, modified the AXFormOpen(Sender) function. Modified the CallCloseOpportunityView(Status), CallUpdateOppCurrency, LoadSnapShot, ppeResellerPopup(Sender), CallOppCalendarSelectionView, pklReasonPopup(Sender), pklTypePopup(Sender), and pklProbabilityExitControl(Sender) subs. Modified for localization. • (6.2.2) For the pklStatus:TPickList control, set HideButtonIfReadOnly to True. Added the pnlSnapShot:TPanel control. Modified for localization. In the script on the form, modified the AXFormOpen(Sender) function. Modified the CallUpdateOppCurrency sub. • (6.2.3) In the script on the form, modified the AXFormChange(Sender), pklStatusPopupReturn(Sender), CallCloseOpportunityView(Status), and LoadSnapShot subs. Modified for localization. • (6.2.4) Modified for localization.	✓	✓	✓	✓
System:Opportunity Reseller • (6.2.1) Modified for localization. • (6.2.2) Modified for localization.	✓	✓		

Form Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
System:Opportunity Statistics • (6.2.3.4) In the script on the form, modified the LoadStatistics(strGroupSQL) sub and added the AXFormClose(Sender) sub.				✓
System:Potential Matches • (6.2.1) Modified for localization. • (6.2.3) In the script on the form, modified the DefineContactsGrid and DefineAccountsGrid functions. Modified the cmdUseAccMatchClick(Sender) and cmdShowContactClick(Sender) subs.	✓		✓	
System:Product Lookup • No changes.	✓			
System:Select Contact(s) • Modified for localization.		✓		
System:SLX_Account_Products_Popup • Added the ProductID Memo object, set DataPaths to Text:V:ACCOUNTPRODUCT:PRODUCTID>PRODUCTID.PRODUCT!PRODUCTID:, Visible to False, and WhenChange to Basic System:SLX_Opportunity_Associate_Product_Set_Description.	✓			
System:SLX_History_Attachments_ManView • For the MainPane object, set HelpContext to 6331.			✓	
System:SLX Manage Hierarchy • Modified for localization.		✓		
System:Ticket Detail • (6.2.1) In the script on the form, included the System:ACI Support script. Added the Dim strCrntACIValue. Modified the AXFormChange(Sender), cmdCopyClick(Sender), FunctionLookupClick(Sender), SetAssignedToPhoneEmail, lveContractClick(Sender), OnACIChange(Sender), OnACIEnterControl(Sender), OnACIExitControl(Sender), OnACIPopup(Sender), OnACIPopupReturn(Sender), lveAccountClick(Sender), lveContactClick(Sender), chkAddtoSearchClick(Sender), lveAccountPopupReturn(Sender), lveContactPopup(Sender), lveContactPopupReturn(Sender), pklStatusChange(Sender), lnkAccountServiceClick(Sender), OnEmailButtonClick(Sender), and AXFormClose(Sender) subs. • (6.2.3) For the dteAssignedDate: TDateTimeEdit control, set DateTime to Ticket:Assigneddate and Text to m/dd/yyyy 00.00 PM. For the lveAssignedTo: TLookupEdit control, set LookupMode to lmOwner. • (6.2.4) Modified for localization.	✓	✓	✓	✓
System:Ticket QBE Screen • For the lveContract:TLookupEdit control, set Lookup to CONTRACT:Referenceno.	✓			
System:Update Multiple Opportunities • (6.2.1) In the script on the form, modified to check for Null and also blank out the date field if the value is 1899. Expanded the Notes column to allow for more data. Added col.CanDeleteText to the creation of the data grid at runtime. • (6.2.2) Modified for localization. • (6.2.4) Modified for localization.	✓	✓		✓

Form Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
System:Update Opportunity Currency • (6.2.1) For frmUpdateOppCurrency:TAXForm, removed the OnCreate event and set the OnOpen event to AXFormOpen. For the txtExchangeRate:TEdit control, set the OnKeyPress event to txtExchangeRateKeyPress. In the script on the form, modified the LoadControls(Mode), lveChangeToPopupReturn(Sender), and txtExchangeRateExitControl(Sender) subs. Added the textExchangeRateChange(Sender) and txtExchangeRateKeyPress(Sender,ByRef Key) subs. • (6.2.2) In the script on the form, modified the LoadControls(Mode), and lveChangeToPopupReturn(Sender) subs. Removed the txtExchangeRateChange(Sender) sub. • (6.2.3) For the In the script on the form, modified the LoadControls(Mode) sub.	✓	✓	✓	
Ticket:Attachments • (6.2.1) Added new control named Button1:TButton. For frmAttach:TAXForm, set the OnFileDragOver event to AXFormFileDragOver and the OnFileDrop event to AXFormFileDrop. For the grdAttach:TDataGrid control, set the Attachment column to no sort and the Last Modified Date column to csDown sort. In the script on the form, added the AXFormFileDragOver(Sender, X, Y) function and the AXFormFileDrop(Sender, FileName, X, Y) sub. • (6.2.4) In the script on the form, enabled/disabled E-mail and Printer menu items based on user options and settings (in Dim blnEmailType and DimblnPrinter, the AXFormOpen(Sender) sub routine, BuildEmailTemplateMenuItems sub routine, and EnableMenuItems sub routine). Added menu security for Attachments tabs.	✓			✓
Ticket:Comments • In the script on the form, modified the memCommentsMouseDown(Sender, Button,X,Y) sub.	✓			
Ticket:Defects • For frmDefects:TAXForm, removed the OnOpen event. For the grdDefects:TDataGrid control, removed the OnClick and OnPopupMenuClick events. In the script on the form, removed the AXFormChange(Sender) sub.	✓			
Ticket:Details • In the script on the form, modified the memDescriptionMouseDown(Sender,Button,X,Y) and memResolutionMouseDown(Sender,Button,X,Y) subs.	✓			
Ticket:Journal • Modified for localization.				✓
Ticket:Support Ticket • For the leTassignedto object, removed the script from the WhenChangeCommit property. Modified the logic to only send e-mail when Status or AssignTo user has changed (based on Account options).	✓			

Changes to Existing Scripts

A check mark indicates the version number in which the change was initially released.

Script Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
Campaign Management:SLX_Add_Edit_Campaign_Response_When_Open • Modified for localization.		✓		

Script Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
System:ACI Support • Added code to update the Assigned Date based on if the AssignedTo user has changed.	✓			
System:Export_Grid_RS_To_Excel • Created new function which can be called to set a variable on whether or not the currency symbol should be displayed for the appropriate data types.	✓			
System:Global Contacts • Added OnAfterFunctionExecute_InsertContactAccount function. Added code to OnAfterInsertContactAccount event handler to ensure there is a valid main view object before working on it.	✓			
System:Global SpeedSearch • Added more meaningful error message to OnBeforeExecuteSpeedSearch event handler. Also, always return True for this Custom Sec Function to avoid unknown function error.	✓			
System:Insert Opportunity Common • (6.2.2) Removed col.FormatType = ftInteger (line 150). Added localized call (line 256). Added the intrinsic function “Round” to the DoGridCalculations function and to several recordset values for quantity (lines 299, 301, and 304).	✓	✓		
System:Insert Ticket • Modified for localization.		✓		
System:Notes History Common • (6.2.1) Modified the logic to display attachments based on record count. If 1, display attachment. If greater than 1, display select attachment view. If Else, display the “no attachment” message. Changed CommonCheckBoxChange procedure to leave the default sort on, which displays a column arrow. • (6.2.2) Modified for localization. • (6.2.4) Modified the sub routine PopupMenuClick for localization. Changed the Select Case to use Item.MenuIndex instead of Item.Caption.	✓	✓		✓
System:OnCompletedActivity • Modified so that the script copies data from the LongNotes field in the Activity table into the ActivityDesc field in the TicketActivity table.	✓			
System:Opportunity Management • (6.2.2) Modified for localization. • (6.2.3.4) Modified the OnBeforeFunctionExecute_ViewOpportunityStatistics function to set global to be read by the SLX OnOpen Opportunity Statistics Report. The report is used to set filtering on the Opportunity Statistics Report. • (6.2.4) Changed the line in the first IF statement to Application.MainViews.ActiveView.GroupsPane.Selection.Item(i). This change will return the correct records that are selected when calling a group.		✓		✓
System:SLX_Add_Alternate_Addresses_AfterDelete	✓			
System:SLX_Add_Alternate_Addresses_BeforeDelete	✓			
System:SLX Address Common • (6.2.2) Modified the BuildAddressPopupMenu script to take an AddressID as a parameter instead of the Address Condition. This allows evaluation of different addresses with the same description, but different addresses, in the same popup menu. • (6.2.4) Updated the BuildAddressPopupMenu method to ignore the check for the Address Description.		✓		✓

Script Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
System:SLX_Common • (6.2.1) Added missing function CheckForNullEmpty. Added GetSQLConditionClauseEx and AddCommaDilimatedTextToWhereClause functions. Added the Sub function AddPicklistItemsToComboBox(strPicklistName, objComboBox, blnAddBlank). Changed the GetSQLConditionClause function. • (6.2.2) Added SLXFormatDateTime function (lines 440 through 444). • (6.2.3) Modified the AddToWhereClause(Field, Clause, Where) function.	✓	✓	✓	
System:SLX_Export_Group_To_Excel • Changed case statement to check for data types ftCurrency and ftFixed (line 230). If found, then check for a currency format type.	✓			
System:SLX Lookup Support • (6.2.1) Modified NavigateToMainViewFromLookup to show the selected record as part of the Lookup Results group using ShowIDsAsLookupResult on the MainView object. Changed query to use the NameLF calculated field. Correction to the GetContactDisplayNameFromID function.	✓		✓	
System:SLX_Manage_Alternate_Addresses_SetShipping • Added Execute SQL statement to update the Entity the dialog originates from with the proper shipping address.	✓			
System:Spell Check • (6.2.1) Changed code to only add the Username/Datestamp in a memo field if the field is not read only. • (6.2.3) Changed the message in the MsgBox from “The spelling check is complete.” to “The spell check is complete.”	✓		✓	
Ticket:META Ticket Account Default Assignment • Rewrote the script to duplicate the same functionality that is in the SalesLogix Client.	✓			
Ticket:META Ticket Activity Charge Builder • Modified to update the Contracts remaining value when changing the Charge Type.		✓		
Ticket:META_Ticket_Assign_To_Validate • Commented out the logic which validated the SecCodeID with that of the current Account’s SecCodeID when SecCodeType is a team. Security is only performed when SecCodeType is a user. Modified the logic to only send e-mail when Status or AssignTo user has changed (based on Account options).	✓			
Ticket:META Ticket PunchInPunchOut • Duplicate pick lists exist causing sub-select query to fail. Changed the query to use a join rather than a sub-query.	✓			
Ticket:META Ticket Punch Out On Close • When the Support Ticket view closes, the script validates that the TicketID exists and performs e-mail notification. Modified the logic to only send e-mail when Status or AssignTo user has changed (based on Account options).	✓			
Ticket:META Ticket Ticketid When Change	✓			
Ticket:META Ticket Update Contract Remaining • Modified to update the Contracts remaining value when changing the Charge Type.		✓		
Ticket:META Ticket Urgency When Exit • The event to populate the NeededBy date has been fixed to pass the correct UrgencyID when creating a new ticket. Change the AddMinutesToDate function to use the SalesLogix internal AddMinutesToDate function and changed parameters to cast the correct type.		✓		

Script Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
Ticket:META Ticket When Open • Modified the logic to only send e-mail when Status or AssignTo user has changed (based on Account options).	✓			
Ticket:SLX Ticket Common • (6.2.1)Added logic to check if AssignedTo user has access to the current account. • (6.2.2) In Sub CheckForExpiredContract(Sender), removed the check for NotIsNull and added NullToZero on aFields(s). • (6.2.4) Removed the localized calls for changing the field background color of In Process and Open tickets (in the OnStatusChange(Sender) sub routine).	✓	✓		✓
Ticket:Support Ticket Memos WhenOpen • Changed query to join to the PickList table to ensure open ticket activities are not of type “timed”. If activities are timed, disable memos. If not timed, leave memos enabled.	✓			
Ticket:Ticket Activity Support • Added logic to the UpdateContractRemaining procedure to convert Null values to 0 (lines 351, 357, and 363). Modified for localization. Modified the GetTimeUnitsFromPrimaryServer function (result = 1/status check). • Added localize calls for the UpdateContract sub routine for the strings in the case statement.		✓		✓

Changes to Existing Templates

A check mark indicates the version number in which the change was initially released.

Template Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
Letter:Base Letter • Changed the alignment on several elements to Left (used to be Justified).		✓		

Changes to Existing Groups

A check mark indicates the version number in which the change was initially released.

Group Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
TICKET:All Open	✓			
TICKET:Follow-Up • Changed the Completeddate column to Completed Date.	✓			
TICKET:My Tickets	✓			
TICKET:Overdue	✓			
TICKET:Punched In	✓			
TICKET:Unassigned	✓			

Changes to Existing Menus

A check mark indicates the version number in which the change was initially released.

Menu Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
System:Standard Menus • Renamed TicketID lookup menu item to Ticket ID.	✓			

Changes to Existing Reports

A check mark indicates the version number in which the change was initially released.

Report Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
Opportunity:Competitor By Opportunity Status • Set the ReportAddCondition OnOpen event to the Opportunity:SLX OnOpen Opportunity Statistic Report Active Script.				✓
Opportunity:Forecast by Account Manager • (6.2.4) Set the ReportAddCondition OnOpen event to the Opportunity:SLX OnOpen Opportunity Statistic Report Active Script.	✓	✓		✓
Opportunity:Products by Opportunity - Sample • Removed all references to ProductPriceLevel table (deprecated in v6.2) and replaced with appropriate fields from Opportunity_Product table			✓	
Opportunity:Quota Vs Actual Sales • (6.2.0.4) Set the ReportAddCondition OnOpen event to the Opportunity:SLX OnOpen Opportunity Statistic Report Active Script.				✓

Changes to Existing Joins

A check mark indicates the version number in which the change was initially released.

Join / Change	6.2.1	6.2.2	6.2.3	6.2.4
ACCOUNTPRODUCT/PRODUCTID to PRODUCT/PRODUCTID • Modified to ignore cascade deletes.			✓	
DEFECTPRODUCT/PRODUCTID to PRODUCT/PRODUCTID • Modified to ignore cascade deletes.			✓	
OPPORTUNITY_PRODUCT/PRODUCTID to PRODUCT/PRODUCTID • Modified to ignore cascade deletes.			✓	
PRODUCTPROGRAM/PRODUCTID to PRODUCT/PRODUCTID • Modified to delete cascade deletes.			✓	
TICKETACTIVITYITEM/PRODUCTID to PRODUCT/PRODUCTID • Modified to ignore cascade deletes.			✓	

Changes to the Advanced Bundle

In v6.2.4 the plugins originally released in the Sales Orders bundle have been moved into the Advanced bundle. These plugins include forms, pick lists, and schema changes.

Schema Changes

The following tables are created when installing the Advanced bundle. A check mark indicates the version in which the schema change was initially released.

New Tables	6.2.1	6.2.2	6.2.3	6.2.4
SALESORDER	✓			
SALESORDERITEMS	✓			

The following fields are created when installing the Advanced bundle.

Table	New Field	6.2.1	6.2.2	6.2.3	6.2.4
ACCOUNTPRODUCT	SALESORDERID	✓			

SQL Statements

The following SQL statements are run when installing the Advanced bundle. A check mark indicates the version in which the statement was initially released.

SQL Statement	6.2.1	6.2.2	6.2.3	6.2.4
DELETE FROM SLXWEBACTION WHERE BUNDLE LIKE 'SalesProcess%' AND NAME IN ('spcompletestep', 'spgetcurrentstatus', 'spaddactivitybyxml', 'spinitsalesprocess', 'spdocontactprocess', 'spdolitrequest', 'spdowebaction', 'spdowebform', 'spgetactionxml')	✓			
SQL Result Set on SLXWEBACTION	✓			

New Forms

A check mark indicates the version number in which the form was initially released.

Form Name	6.2.1	6.2.2	6.2.3	6.2.4
Opportunity:Order Details	✓			

New Scripts

A check mark indicates the version number in which the script was initially released.

Form Name	6.2.1	6.2.2	6.2.3	6.2.4
System:SP_InitSalesProcess • Called from the Insert Opportunity form.				✓

Changes to Existing Forms

A check mark indicates the version number in which the change was initially released.

Form Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
Opportunity:Sales Process • (6.2.1) The grdSalesProcessDateColumnValidateInput function was returning a boolean value which is incorrect. In the script on the form, changed the code to revert back to the previous date if the value entered is not a valid date. • (6.2.2) For the Form:TAXForm object, set Height to 411. For the grdSalesProcess:TDataGrid object, set Height to 327. For the pnlGrid:TPanel object, set Height to 327. • (6.2.4) Added a localize call to the BuildComboBox sub.	✓	✓		✓
System:Add Edit Sales Order • (6.2.1) Added the ActualID field to the SalesOrderItems table. The ActualID value is obtained from the Product table. • (6.2.2) In the script on the form, changed lblBillDescription.Caption to txtBillAddressID.Text for sub cmdBillAddressClick(Sender) and sub cmdShipAddressClick(Sender).	✓	✓		
System:Opportunity Statistics • (6.2.1) Added the SLXIsNumeric function to check the value of the Opportunity.SalesPotential, Opportunity.ActualAmount, and Opportunity.WeightedTotal fields for upgrades from databases earlier than 5.0. Added logic to check for the year to be greater than 1900 (so an opportunity without an estimated close date displays properly). Added an additional check for the new MainView.GroupsPane.IDs collection. If the GroupID is empty, the user is on the Lookup Results group. • (6.2.0.4) Set the Global Variable in the Global Script that contains a list of Opportunity IDs to be shown in Opportunity Statistics View. Use that list as an IN clause on the Opportunity Statistics reports.	✓			✓
System:SP_Update Stages • In the script on the form, modified the SetCompletedFlag function to do an ADO update instead of a SQL2 update.	✓			
System:Update Multiple Opportunities • In the script on the form, added code to check for Null and also blank out the date field if the value is 1899. Expanded the Notes column to allow for more data. Added col.CanDeleteText to the creation of the data grid at runtime. The user can now use the backspace and delete keys. • Added localized calls to the cboSelectedUpdate sub routine.	✓			✓

Changes to Existing Scripts

A check mark indicates the version number in which the change was initially released.

Script Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
System:SP_InitSalesProcess • Modified for localization.				✓
System:SP_SalesProcessFunctions • (6.2.1) Modified so the objCon is properly dimmed within the GetAccManagerID method. • (6.2.4) Modified for localization.	✓			✓

Changes to Existing Pick Lists

A check mark indicates the version number in which the change was initially released.

Pick List Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
Sales Order Status	✓			
Sales Order Type	✓			

Changes to Existing Reports

A check mark indicates the version number in which the change was initially released.

Report Name / Change	6.2.1	6.2.2	6.2.3	6.2.4
Opportunity:Sales Process Stage Analysis • Set the ReportAddCondition OnOpen event to the Opportunity:SLX OnOpen Opportunity Statistic Report Active Script.				✓
Opportunity:Sales Process Step Usage • Set the ReportAddCondition OnOpen event to the Opportunity:SLX OnOpen Opportunity Statistic Report Active Script.				✓