Applying Service Pack 3 for SalesLogix Version 6.2

Version 6.2.3

Developed by SalesLogix Technical Publications

insights for the life of your business™

Applying Service Pack 3 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 SalesLogix.techpubs@sage.com.

Copyright

Version

Version 6.2.3 (929)
September 2005

Printed in the United States of America.

Trademarks

SalesLogix is a registered trademark of Best Software, Inc.
Other product names may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.

Disclaimer

Best Software has thoroughly reviewed this manual. All statements, technical information, and recommendations in this manual and in any guides or related documents are believed reliable, but the accuracy and completeness thereof are not guaranteed or warranted, and they are not intended to be, nor should they be understood to be, representations or warranties concerning the products described. Best Software assumes no responsibility or liability for errors or inaccuracies with respect to this publication or usage of information. Further, Best Software reserves the right to make changes to the information described in this manual at any time without notice and without obligation to notify any person of such changes.

Technical Support

Technical Support is available to customers with support contracts directly from Best 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/Best Online Support and Services Web site.

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

Revisions to this book are posted on the SalesLogix SupportOnline/Best 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 3 for SalesLogix version 6.2. This service pack includes defect 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 6.2.3, 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 3. 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/Best Online Support and Services Web site (http://support.saleslogix.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 3 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** Do not install this service pack if you are currently running SalesLogix on Oracle 8. In version 6.2.3, support for Oracle 8 is discontinued. The functionality in this service pack will not run on Oracle version 8 or earlier.	
	2 Install SalesLogix v6.2.3 in a test environment before installing on your production database. **Note** Refer to SupportOnline/Best 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 v6.2.3 plugins. However, this bundle can be used as a backup of your customizations.	

✓	Task	Notes
	4 Understand the changes in v6.2.3. **Note** This service pack also includes the changes released in v6.2.2 and v6.2.1. These changes are detailed in Tasks 5 and 6.	

Task 4 Details

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

Virtual Server-Side Cursors

Virtual Server-Side Cursors have been 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 improves large group performance and memory usage in the SalesLogix Client Groups and List views. This includes 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 eliminates the need for the SalesLogix Client to manually set the GlobalID in script for new tables and to simplify 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 - 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
	5 Understand the changes in v6.2.2. **Note** Service Pack 3 is a cumulative release. Therefore, the changes previously released in v6.2.2 are also included in v6.2.3.	

Task 5 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 recommended on multi-user systems such as Citrix 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 28.

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/Best 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/Best Online Support and Services.

✓	Task	Notes
	6 Understand the changes in v6.2.1. **Note** Service Pack 3 is a cumulative release. Therefore, the changes previously released in v6.2.1 are also included in v6.2.3.	

Task 6 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 v6.2.3 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

Service Pack 1 included a new feature for synchronization. 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.

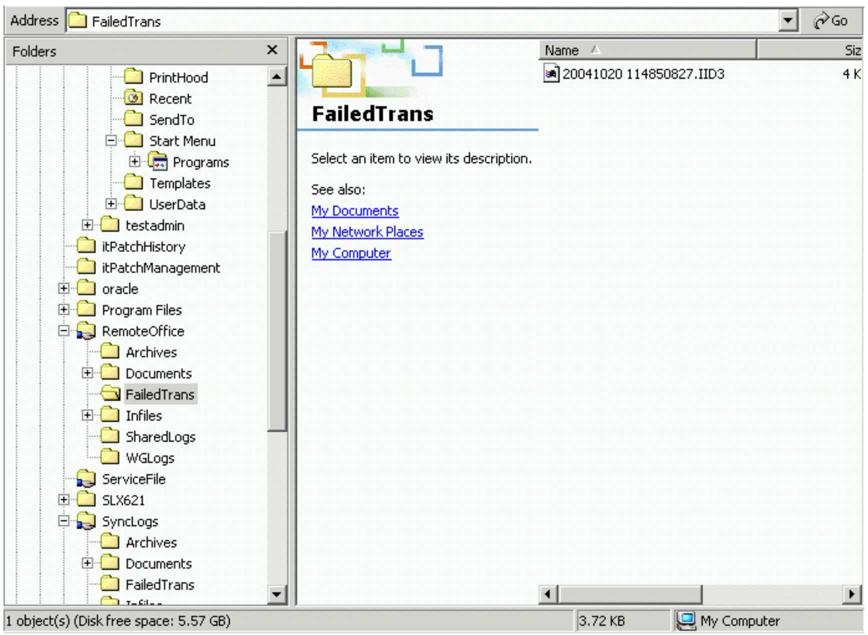
✓	Task	Notes

Task 6 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" will 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 6 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.	
	7 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 3 bundle.	
	8 After reviewing the plugin changes, determine if you are going to add your customizations to the v6.2.3 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 retrofit your existing plugins with the localization changes.	
	9 If you currently have the Advanced bundle applied to your system, review “Changes to the Advanced Bundle” on page 29 to determine if you have customized any plugins that are included in the SalesLogix Advanced v6.2 Service Pack 3 bundle. Determine if you are going to add your customizations to the v6.2.3 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 retrofit your existing plugins with the localization changes.	
	10 If you have customized any of the Web components in a separate package, locate the package folder and make a backup copy. For example, if you previously created a package called MyCustomizations, copy the entire MyCustomizations folder to a backup location.	

✓	Task	Notes
	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.	
	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 Back up your production database. Verify the backup is successful and complete.	

✓	Task	Notes
	18 Run the SLX_v62_SP3 installation on your SalesLogix Server.	

Task 18 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_SP3.exe**.
2. On the **SalesLogix v6.2 Service Pack 3** screen, select your installation method:
   - **Install the SalesLogix v6.2 Service Pack 3** 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 3** 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 34.
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 of the patch wizard. If you created automated installations of the SalesLogix Network Client and/or Support Client, you can upgrade the installations to Service Pack 3 using this option (you must browse out to the .msi files and click Update). Once the images are updated, they must be distributed to all SalesLogix Network Client and/or Support Client users.
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
	19 Run the SLX_v62_SP3 installation on your Administrative Workstation.	
	20 Start the SalesLogix Administrator.	

21 Apply the Service Pack bundle named SalesLogix v6.2 Service Pack 3.sxb.

Task 21 Details

If you extracted the service pack files to your local computer (task 18), 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_SP3.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 3.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.
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 v6.2.3 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**.

22 If your installation includes the Advanced bundle, apply the bundle named SalesLogix Advanced v6.2 Service Pack 3.sxb.

Task 22 Details

While the bundle is installed, click Yes or Yes to All on any confirmation message boxes for overwriting system plugins. Disregard any messages regarding table indexes, and click OK.

23 Apply the v6.2.3 changes listed in Chapter 2, “Changes in This Release” You can use one of the following strategies:

- Add your customizations to the v6.2.3 plugins (recommended).
- Add the v6.2.3 changes to your custom plugins.

Task 23 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 6.2.3.

24 After applying 6.2.3 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.

✓	Task	Notes
	25 Run the SLX_v62_SP3 installation on your SpeedSearch Server.	
	26 Run the SLX_v62_SP3 installation on each of your SalesLogix Network user's computers.	
	27 Run the SLX_v62_SP3 installation on each of your Support user's computers.	
	28 If necessary, install the SLXSystem.exe as a Windows Service. **Note** For more information, see task 5. **Task 28 Details** The SLXSystem service starts and stops when connections are created. However, by default, the SLXSystem.exe continues to shut down after a minute. 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 design, the SLXSystem.exe service is set to Manual. This allows the service to shut down one minute (or the value set in the registry) after it detects there are no longer any connections. **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)	
	29 Run the SLX_v62_SP3 installation on your Synchronization Server(s). **Task 29 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></td><td>30 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></td><td colspan="2">Task 30 Details
FailedTrans information is detailed in task 6.

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>31 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></td><td>32 Update your Synchronization Service jobs.</td><td></td></tr>
<tr><td></td><td colspan="2">Task 32 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>33 Restart the Sync Service(s).</td><td></td></tr>
</table>

✓	Task	Notes
	Upgrade SalesLogix Remote Users/Offices	
	34 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. **Task 34 Details** To upgrade Remote Users, you can distribute the SalesLogix Client 6.2 SP3.msp file rather than the entire Service Pack patch. This file is smaller and upgrades only the necessary SalesLogix Client pieces. The SalesLogix Client 6.2 SP3.msp file is located in the folder you specified when extracting the service pack files (see task 18). After receiving the file, instruct the Remote User to double-click the SalesLogix Client 6.2 SP3.msp file.	
	35 Direct Remote Users and Offices to run the SLX_v62_SP3 installation (or if appropriate, the SalesLogix Client 6.2 SP3.msp file) on their remote computers.	
	36 Direct Remote Users to start the SalesLogix Client.	
	37 Instruct Remote Users and Offices to sync immediately after upgrading.	
	Upgrade Web Components (if applicable)	
	38 If your SalesLogix implementation includes Web Components, refer to the *Upgrading SalesLogix Web Host_Client to 623.pdf* for upgrade instructions.	
	Perform Post-Upgrade Tasks	
	39 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 39 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.	
	40 Instruct users to log on to SalesLogix.	
	41 If necessary, in the location where you extracted the service pack (workplan task 18), 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, and 6, this service pack fixes numerous product defects. This includes defects previously addressed in hot fixes.

Refer to the Fixed Issues List for v6.2.3 on SupportOnline/Best 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
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 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'		✓	
update Resynctabledefs set omnidirectional = 'X' where tablename = 'SyncSequencing'		✓	
update Resynctabledefs set secondaryfield = 'TARGETSITE' where tablename = 'SyncSequencing' and secondaryfield is null		✓	
UPDATE SECTABLEDEFS SET DATETIMETYPE = ‘D’ WHERE TABLENAME = ‘CAMPAIGN’ AND FIELDNAME = ‘ACTUALLAUNCHDATE’ • Added to correctly display the date value in the Web Client.	✓		

Schema Change	6.2.1	6.2.2	6.2.3
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 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 Upgrade bundle.

New Table	6.2.1	6.2.2	6.2.3
SYNCSECTABLES			✓

The following fields are created when installing the Upgrade bundle.

Table	New Field	6.2.1	6.2.2	6.2.3
ACCOUNTPRODUCT	OPPORTUNITYID, OPPPRODUCTID, QUANTITY		✓	
BRANCHOPTIONS	LICNUM	✓		
RESYNCTABLEDEFS	DEFAULTGLOBALIDPATH • Reserved for the v6.2.3 Feature Pack for Marketing.			✓
SECTABLEDEFS	AUTOINCREMENT	✓		
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
System:Add Edit Address	✓		

Form Name	6.2.1	6.2.2	6.2.3
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
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	✓		
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
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
System:Account Details • For the mvAccountDetail:TMainAXForm, set OnSwitchViewMode to MainAXFormSwitchViewMode. Included the SLX Address Common script on the form. Removed the code that was setting the ActiveControl property. This causes the F1 hot key to fail until focus is set to the Detail view. Changed menu item to call MainView:AdWriteEmailtoSelectedRecords sec function. Changed the Height of the MainView to 414. Updated the AXFormChange event to account for possible null values when setting the current address option. (6.2.1) • In the script on the form, added the OnFunctionEnabled_ViewAltAddress procedure to disable the View Alternate Address menu item when view is read-only (lines 3-9). (6.2.2)	✓	✓	
System:Contact Details • Removed the code that was setting the ActiveControl property. This causes the F1 hot key to fail until focus is set to the Detail view. Changed menu item to call MainView:AdWriteEmailtoSelectedRecords sec function. Resized the view. (6.2.1) • In the script on the form, added the OnFunctionEnabled_ViewAltAddress procedure to disable the View Alternate Address menu item when view is read-only (lines 11-17). (6.2.2)	✓	✓	
System:Opportunity Details • Removed the code that was setting the ActiveControl property. This causes the F1 hot key to fail until focus is set to the Detail view. Changed menu item to call MainView:AdWriteEmailtoSelectedRecords sec function. Changed two lines in the “Send Email” subroutine. Added the call Application.BasicFunctions.SystemInfoFor(“BaseCurrency”) to get the base currency required for the labels.	✓		
System:Ticket Details • Removed the code that was setting the ActiveControl property. This causes the F1 hot key to fail until focus is set to the Detail view. Changed menu item to call MainView:AdWriteEmailtoSelectedRecords sec function. Added code to update the Assigned Date based on if the AssignedTo user has changed.	✓		

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
Account:Contracts • For the frmContractAccounts:TAXForm object, set Height to 564 and Width to 507. For the grdContractAcount:TDataGrid object, set Height to 550. For the splGrid:TSplitterPanel object, set Height to 552 and Width to 495. In the script on the form changed the data type assigned to the grid to String (line 175). Added a condition when the “Perpetual” contract type is selected to disable the Quantity field. Calculation was failing when attempting to perform a DateDiff on a null value. Changed to check for a null value before calculating the date.	✓		
Account:Notes-History • For the cmdOptionsFull:TButton object, set Top to 2.	✓		

Form Name / Change	6.2.1	6.2.2	6.2.3
Account:Opportunities • In the script on the form, wrapped the SLXIsNumeric function call around the result. Added NullToZero function in the UpdateStatusBarTotals. 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. This is done in the function grdAccountOppsFormatColumnText. (6.2.1) • Modified for localization. (6.2.2) • For the frmOpportunities:TAXForm, set Height to 617 and Width to 481. For the grdAccountOpps:TDataGrid object, set Height to 584 and Width to 461. For the txtTotals:TEdit object, set Width to 469. (6.2.3)	✓	✓	✓
Account:Reseller Opportunities • Modified for localization. (6.2.1) • Added the grid event grdOpportunitiesFormatColumnText to handle the Days Open field. Also altered the SQL statement to include the new fields OpportunityID, DateOpened, ActualClose, and CreateDate that were added to the grid as not visible so calculations can be done correctly for Days Open. (6.2.2)	✓	✓	
Account:Tickets • The wrong calculated field was being used (FullName). Changed to LastFirst.	✓		
Contact:Notes-History • Changed the Word icon to be consistent with other Notes-History views.	✓		
Contact:Opportunities • Display name from Contact Details view was being used which displays First Last. Changed to call GetContactDisplayNameFromID which returns the name formatted Last First. Added a function to calculate the days open. 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. This is done in the function grdAccountOppsFormatColumnText.	✓		
Contact:Tickets • Fixed portion of a script that was referencing the wrong object.	✓		
Opportunity:Competitors • Modified for localization. (6.2.2) • Modified the script on the form so that setting the RowSelect property will essentially make the grid Read Only and will disable the DisableEditor property for all columns. (6.2.3)		✓	✓
Opportunity:Contacts • In the script on the form, modified to set a maximum count of 250 (line 159). (6.2.1) • Modified the script on the form so that setting the RowSelect property will essentially make the grid Read Only and will disable the DisableEditor property for all columns. (6.2.3)	✓		✓
Opportunity:Notes-History • For the memNotes:TMemo object, set Width to 2. (6.2.1) • Modified for localization. (6.2.2)	✓	✓	
Opportunity:Order Details • Modified for localization.		✓	

Form Name / Change	6.2.1	6.2.2	6.2.3
Opportunity:Products • Modified the script on the form (line 366). (6.2.1) • Added the intrinsic function “Round” to the DoGridCalculations function. Also added Round to several recordset values for Quantity. Modified for localization. (6.2.2) • Modified the script on the form so that setting the RowSelect property will essentially make the grid Read Only and will disable the DisableEditor property for all columns. (6.2.3)	✓	✓	✓
System:Account Detail • For the DummyLookup:TLookupEdit object, set Left to 317 and Top to 97. For the frmAccountDetail:TAXForm form, set Height to 179. 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, added the SLX Address Common script. (6.2.1) • Removed some old code in the FormChange event handler that attempted to disable a menu item when security was view / read-only. In the script on the form, modified to determine if the account is Read Only for the Current User (lines 200-202 and 253). In the script on the form, modified the BuildAddressPopupMenu script within SLX Address Common 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. Added localized calls around the picklist.text properties. Also added calls around the “if” conditions so that when data is stored in French/German, the pick list value will be compared to the localized string. (6.2.2)	✓	✓	
System:AddAddressDialog • In the script on the form, modified the Update Address procedure to check the origin of the request before prompting (line 1210).	✓		
System:Add Center • Updated for localization (changed EditBoxes to Label controls and replaced Panels with Bevels).		✓	
System:Add Edit Competitor • In the script on the form, converted a NULL value to an empty string (line 51). Opening a competitor without a name will force the user to enter a name prior to saving. For the memWeakness:TMemo object, set MaxLength to 128. (6.2.1) • Modified for localization. (6.2.2)	✓	✓	
System:Add Edit Product Package • For the grdMembers object, set OnEdited to grdMembersEdited. Modified the script on the form to separate SQL based on whether the packages are being editing or created. If created, a separate insert is done to create the package record.	✓		
System:Add Edit Ticket Activity • For the memComments: TMemo object, removed the OnClick event. Added a vertical scroll bar to the Memo control. For the cmdComments: TButton object, disabled the Flat property. In the script on the form, removed blnUserDateStamp (line 11 and 22), set type to a double rather than an integer (line 158), and disabled the Spell Check menu item if Word is not installed. (6.2.1) • Added an additional variable to be checked when in edit mode to see if the “Charge Type” was changed. If so, the Contracts remaining value is updated. (6.2.2)	✓	✓	

Form Name / Change	6.2.1	6.2.2	6.2.3
System:Add Edit Ticket Rate • For the txtType:TEdit object, set TabOrder to 0. For the txtDescription:TEdit object, set TabOrder to 1. For the txtAmount:TEdit object, set TabOrder to 2.	✓		
System:Add New Contact Account • For the frmAddNewConAcc: TAXForm object, removed the OnDestroy event. Modified the creation order 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. They are no longer necessary since we are not doing any SQL2 transactions. Changed the Height of the MainView to 414. Added event for the OnExitControl of the txtFax object. If txtAccFax is empty and enabled, the contact's fax number is copied over. Logic to populate the addresses was called multiple times with the last passing an incorrect ID. Changed the procedure to call GetDisplayAddress function and removed the repetitive calls. In the script on the form, included the System:SLX Address Common script and updated the Description and Salutation fields values to 64 (line 725 and 726). (6.2.1) • Added localized calls for caption headings and message boxes. (6.2.2)	✓	✓	
System:Add Opportunity Product • For the ppeExchangeRate:TPopupEdit object, set OnExitControl to ppeExchangeRateExitControl and OnKeyPress to ppeExchangeRateKeyPress. Changed code to change the KeyFieldID field of the product grid RS to the newly created OppProductID after the record is inserted (line 523). Added a call to ReplaceSingleQuote prior to executing the query. Added code to disable the Manage Products button if the user does not have permissions to manage products. Added a function to check if the Opp Currency is the Base Currency. If so, then change the control to ReadOnly so the user cannot manually change the rate of "1". Added an if statement to check for ppeExchangeRate.text being equal to " " and setting it to "1" if blank. (6.2.1) • In the script on the form, changed the logic 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 (starts at line 652). Modified for localization. Removed the CheckForBaseCurrency and CheckForBaseCurrencyPopUp calls. (6.2.2) • In the script on the form, added the CheckForBaseCurrency and CheckForBaseCurrencyPopUp calls back to the script. (6.2.3)	✓	✓	✓
System:ChooseAddContactAccount • In the script on the form, added code for enabling/disabling buttons depending on which radio button is selected (lines 29, 30, 41, and 47). (6.2.2) • In the script on the form, modified to open new contact/account in a new main view window (for the Application.MainViews.AddEx "System:Contact Details" and Application.MainViews.AddEx "System:Account Details" set open existing to False). (6.2.3)		✓	✓
System:Close Opportunity • In the script on the form, 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 (changes start at line 211). • In the script on the form, modified the Sub txtActualAmtExitControl(Sender) to check for a Null value. If the Actual Amount field is blanked out, the sub will place "0" in the field. (6.2.3)	✓		✓

Form Name / Change	6.2.1	6.2.2	6.2.3
System:Contact Detail • For the cmdContactType:TButton object, set TabOrder to 33 and Top to 177. For the frmContactDetail:TAXForm form, set Height to 235. For the mnuAddress:TPopupMenu object, set Left to 514 and Top to 44. Added the txtAddressID:TEdit object; set Height to 10, Left to 550, Name to txtAddressID, TabOrder to 56, Text to Contact:AddressID, Top to 50, Width to 10 and set the OnChange event to txtContactIDChange. For the txtContactID:TEdit object, set Left to 540 and Top to 50. (6.2.1) • In the script on the form, modified to determine if the account is Read Only for the Current User (lines 153 - 155, 202, 204, and 210). In the script on the form, modified the BuildAddressPopupMenu script within SLX Address Common 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. (6.2.2)	✓	✓	
System:Contract Detail • In the script on the form, disabled the Spell Check menu item if Word is not installed. When copying a Contract of type “Days”, the Start and End Date fields as well as Grace Period will be enabled. Added a condition when the “Perpetual” contract type is selected to disable the Quantity field. Disabled the Begin Contract after First Ticket check box when the Perpetual contract type is selected. This check box is only enabled for the contract type “Days”. Added logic to recalculate the List Price based on changing the contract amount. (6.2.1) • In the script on the form, removed the CheckContractType sub procedure (lines 239-254). Modified to check for an empty string the same as the CheckForExpiredContract procedure (line 449 and 457). Modified for localization. (6.2.2)	✓	✓	
System:Copy Quota		✓	
System:Edit Opportunity Competitor • For the memWeakness:TMemo object, set MaxLength to 128.	✓		
System:Edit Opportunity Contact • Removed the e-mail menu as this logic currently does not exist for scheduling activities.	✓		
System:Edit View Notes History • For the memNotes:TMemo object, set ScrollBars to ssVertical and Width to 440. For the cmdCancel:TButton, cmdHelp:TButton, and cmdOK:TButton objects, set Left to 13. For the pnlButtons:TPanel object, set Left to 440 and Width to 98. (6.2.1) • For the Form:TAXForm object, set HelpContext to 1678. (6.2.2)	✓	✓	
System:Insert New Ticket • For the lveAssignedTo:TLookupEdit object, set LookupRestrictField to SECCODETYPE, LookupRestrictOp to <>, and LookupRestrictValue to D. This removes template descriptions from the lookup. In the script on the form, included the System:ACI Support script, rearranged the sequence in which the QueMessage function is called, and added logic to check if AssignedTo user has access to current account. Added logic to populate Account if Account is empty when selecting a Contact. In the script on the form, disabled the Spell Check menu item if Word is not installed, added code to update the Assigned Date based on if the AssignedTo user has changed, and added information to clear a contract if “No” is selected at the “Warning: This Contract has expired...” message (line 645). (6.2.1) • Set LookupRestrictField to SECCODETYPE, LookupRestrictOp to <> and LookupRestrictValue to ‘D’. This removes template descriptions from the lookup. Added localized calls for the SetDefaults sub routine in the Status area. (6.2.2) • For the lveAssignedTo: TLookupEdit object, set LookupMode to lmOwner. (6.2.3)	✓	✓	✓

Form Name / Change	6.2.1	6.2.2	6.2.3
System:Insert New Ticket - Account Service Information • In the script on the form, changed the data type assigned to the grid to String (line 258).	✓		
System:Insert Opportunity • For the lveAccount:TLookupEdit object, removed the OnChange event. For the ppeExchangeRate:TPopupEdit object, set OnExitControl to ppeExchangeRateExitControl and OnKeyPress to ppeExchangeRateKeyPress. Removed a line of code that was setting the AccountID before a case statement that builds the SQL statement based on UserOptions. The AccountID was never getting set to the new value when the user changed the Account Name in the lookup. Added a function to check if the Opp Currency is the Base Currency. If so, then change the control to ReadOnly so the user cannot manually change the rate of "1". Changed the view to use the AccountID for the GetAccountMgr function. Removed the code that inserted an actual amount value when creating a new opportunity. Added an if statement to check for ppeExchangeRate.text being equal to " " and setting it to "1" if blank. Added logic to check if multi-currency is enabled. If not enabled, the ExchangeRate, ExchangeRateDate, ExchangeRateCode, and ExchangeRateLocked fields are not updated in the Opportunity table. For the lveAccount:TLookupEdit object, removed the OnChange event. Added a function to the OnClose event. In the function AXOnCloseQuery, there is a check for a boolean value. The boolean value is used to check if the code is valid before calling the SaveOpportunity subroutine. (6.2.1) • In the script on the form, added the RefreshTabMiddleViews function for the Contact Opportunities view. Added "Left" function to the text description on the SaveOpportunityDetailInfo sub routine. If the value in the text property is greater than 64, the value is truncated. Added the GetAccountMgrForSelectedAccount(strAccountID) call to the lveAccount fields OnPopUpReturn event. This returns the current AccountMgrID for the account selected for the opportunity, not the current account. Modified for localization. Removed the CheckForBaseCurrency and CheckForBaseCurrencyPopUp calls. (6.2.2) • In the script on the form, added the CheckForBaseCurrency and CheckForBaseCurrencyPopUp calls. back to the script (6.2.3)	✓	✓	✓
System:Manage Area Category Issue • In the script on the form, changed the word "To" to lowercase (to) (line 147). Changed the script to only perform the "Application.BasicFunctions.GetMenuSecurity" action if the application is SalesLogix.	✓		
System:Manage Customer Service Defaults • In the script on the form, modified to omit Retired users (line 231). Corrected the spelling of Calendar on the Customer Service Options > User Defaults tab. (6.2.1) • Added localized calls to the grid that is built at runtime. (6.2.2)	✓	✓	
System:Manage Customer Service User Defaults - Individual • For the chkTicketActivity:TCheckBox object, corrected spelling of the word calendar.	✓		
System:Manage Customer Service User Defaults - Multiple • For the chkTicketActivity:TCheckBox object, corrected spelling of the word calendar.	✓		

Form Name / Change	**6.2.1**	**6.2.2**	**6.2.3**
System:Manage Product • Changed the display format of the Price column to omit the currency symbol if multi-currency is enabled. Added error handling around various functions. (6.2.1) • In the script on the form, changed the DeleteSelectedProduct sub function to call CascadeDelete. (6.2.3)	✓		✓
System:Manage Product Packages • In the script on the form, added Application.BasicFunctions.GetIDFor("PACKAGES") (line 44). (6.2.1) • Added localized call to the form caption on the call to Add Edit Product Package. (6.2.2)	✓	✓	
System:Manage Quota • In the script on the form, changed the logic to only enable the Copy button if the selected Sales Rep has a Quota(s). (6.2.2)	✓	✓	
System:Memo PopUp • In the script on the form, disabled the Spell Check menu item if Word is not installed.	✓		
System:Opportunity Defaults • For the cmdProducts:TButton object, set Left to 7, TabOrder to 2, and Top to 283. (6.2.1) • Removed the CheckForBaseCurrency and CheckForBaseCurrencyPopUp calls. (6.2.2)	✓	✓	✓
System:Opportunity Detail • Changed logic to ensure that there is a record selected prior to assigning a value to the Reseller lookup. Added a function to check if the Opp Currency is the Base Currency. If so, then change the control to ReadOnly so the user cannot manually change the rate of "1". For the pklProbability:TPickList object, set OnEnterControl to pklProbabilityEnterControl. Added code to the snapshot to display a new default message. The global variable for "mode" was not being set for an existing opportunity. Set the global variable to get the exchange rate from the Opportunity table. Added a condition to check for a blank value for the Probability. If the probability is empty, the Probability field is set to 0. (6.2.1) • For the pklStatus:TPickList object, set HideButtonIfReadOnly to True. Added the pnlSnapShot:TPanel object; set Height to 209, Left to 336, TabOrder to 16, Top to 10, and Width to 447. Modified for localization. (6.2.2) • In the script on the form, modified the pklStatusPopupReturn(Sender) and CallCloseOpportunityView(Status) sub function for localization. Added the function CheckForNull and NullToZero to the CallCloseOpportunityView sub function. Set the strStatus variable so the Actual Amount field is populated correctly when an opportunity is Closed-Won. Removed a single quote from the SendEmail sub routine. (6.2.3)	✓	✓	✓
System:Opportunity Reseller • Changed logic to ensure that there is a record selected prior to assigning a value to the Reseller lookup. (6.2.1) • Modified for localization. (6.2.2)	✓	✓	
System:Potential Matches • In the script on the form, modified for localization. (6.2.1) • In the script on the form, modified to close and destroy recordsets when applicable. Modified to remove the reference the script was making to an invalid object contained within a With statement. (6.2.3)	✓		✓
System:Product Lookup	✓		

Form Name / Change	6.2.1	6.2.2	6.2.3
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 • For the lveAssignedTo:TLookupEdit object, set LookupRestrictField to SECCODETYPE, LookupRestrictOp to <>, and LookupRestrictValue to D. This removes template descriptions from the lookup. In the script on the form, added logic to check if AssignedTo user has access to current account. Initialized the timer to False from the OnChange event on the form. Updated seccodeid if the account changes. Changed the tab order on the view. (6.2.1) • Set LookupRestrictField to SECCODETYPE, LookupRestrictOp to <> and LookupRestrictValue to 'D'. This removes template descriptions from the lookup. (6.2.2) • For the dteAssignedDate: TDateTimeEdit object, set DateTime to Ticket:Assigneddate. For the lveAssignedTo: TLookupEdit object, set LookupMode to lmOwner. (6.2.3)	✓	✓	✓
System:Ticket QBE Screen • For the lveContract:TLookupEdit object, set Lookup to CONTRACT:Referenceno.	✓		
System:Update Multiple Opportunities • Added localized calls to caption properties for the grid column headings.		✓	
System:Update Opportunity Currency • Added SetMCValue MC_OppExchangeRate, txtExchangeRate.Text to the OK ButtonClick. This was only being set on the OnPopupReturn from the Lookup control. Changed for when users alter the rate manually. Added logic and globals to check values before and after the user enters and exits the view. Modified to check for blank values for the exchange rate. For the txtExchangeRate:TEdit object, set OnExitControl to txtExchangeRateExitControl and OnKeyPress to txtExchangeRateKeyPress. Added a function to check if the Opp Currency is the Base Currency. If so, then change the control to ReadOnly so the user cannot manually change the rate of "1". (6.2.1) • Modified for localization. In the script on the form, added logic to determine if the user can update the exchange rate. (6.2.2) • In the script on the form, modified to CheckForBaseCurrency(lveChangeTo.Text). (6.2.3)	✓	✓	✓
Ticket:Attachments • Added new object named Button1:TButton; set the OnClick event to Button1Click. For the frmAttach:TAXForm form, set OnFileDragOver to AXFormFileDragOver and OnFileDrop to AXFormFileDrop. Added a call to refresh the grid after inserting a file.	✓		
Ticket:Comments • In the script on the form, disabled the Spell Check menu item if Word is not installed (line 61).	✓		

Form Name / Change	6.2.1	6.2.2	6.2.3
Ticket:Defects • For the frmDefects:TAXForm form, set Width to 288 and OnOpen to blank. For the grdDefects:TDataGrid object, set the Height to 362, and OnClick and OnPopupMenuClick events to blank. For the memDescription object, set Width to 90. For the memResolution:TMemo object, set Height to 212 and Width to 90. For the pnlDescription:TPanel and pnlResolution:TPanel objects, set Width to 90. For the splComments:TSplitterPanel object, set Height to 362 and Width to 92. For the splGrid:TSplitterPanel object, set Height to 364 and Width to 276.	✓		
Ticket:Details • In the script on the form, disabled the Spell Check menu item if Word is not installed (line 127).	✓		
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
Campaign Management:SLX_Add_Edit_Campaign_Response_When_Open • Modified for localization.		✓	
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 • 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). (6.2.2)	✓	✓	
System:Insert Ticket • Modified for localization.		✓	
System:Notes History Common • 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. (6.2.1) • Modified for localization. (6.2.2)	✓	✓	

Script Name / Change	6.2.1	6.2.2	6.2.3
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 • Modified for localization.		✓	
System:SLX_Add_Alternate_Addresses_AfterDelete	✓		
System:SLX_Add_Alternate_Addresses_BeforeDelete	✓		
System:SLX Address Common • 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.		✓	
System:SLX_Common • Added missing function CheckForNullEmpty. Added GetSQLConditionClauseEx and AddCommaDilimatedTextToWhereClause functions. Added the Sub function AddPicklistItemsToComboBox(strPicklistName, objComboBox, blnAddBlank). Changed the GetSQLConditionClause function. (6.2.1) • Added SLXFormatDateTime function (lines 440 through 444). (6.2.2) • Modified the AddToWhereClause(Field, Clause, Where) function. (6.2.3)	✓	✓	✓
System:SLX_Export_Group_To_Excel • Change 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 • 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. (6.2.1)	✓		✓
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 • Changed code to only add the Username/Datestamp in a memo field if the field is not read only. (6.2.1) • Changed the message in the MsgBox from "The spelling check is complete." to "The spell check is complete." (6.2.3)	✓		✓
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.	✓		

Script Name / Change	6.2.1	6.2.2	6.2.3
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.		✓	
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 • Added logic to check if AssignedTo user has access to the current account. (6.2.1) • In Sub CheckForExpiredContract(Sender), removed the check for NotIsNull and added NullToZero on aFields(s). (6.2.2)	✓	✓	
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).		✓	

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
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
TICKET:All Open	✓		
TICKET:Follow-Up • Changed the Completeddate column to Completed Date.	✓		
TICKET:My Tickets	✓		
TICKET:Overdue	✓		

Group Name / Change	6.2.1	6.2.2	6.2.3
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
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
Opportunity:Forecast by Account Manager	✓	✓	
Opportunity:Products by Opportunity - Sample • Removed all references to ProductPriceLevel table (deprecated in v6.2) and replaced with appropriate fields from Opportunity_Product table			✓

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
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 v6.2.3 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
SALESORDER	✓		
SALESORDERITEMS	✓		

The following fields are created when installing the Advanced bundle.

Table	New Field	6.2.1	6.2.2	6.2.3
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
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
Opportunity:Order Details	✓		

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
Opportunity:Sales Process • 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. (6.2.1) • 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. (6.2.2)	✓	✓	

Form Name / Change	6.2.1	6.2.2	6.2.3
System:Add Edit Sales Order • Added the ActualID field to the SalesOrderItems table. The ActualID value is obtained from the Product table. (6.2.1) • In the script on the form, changed lblBillDescription.Caption to txtBillAddressID.Text for sub cmdBillAddressClick(Sender) and sub cmdShipAddressClick(Sender). (6.2.2)	✓	✓	
System:Opportunity Statistics • 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.	✓		
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.	✓		

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
System:SP_SalesProcessFunctions • Modified so the objCon is properly dimmed within the GetAccManagerID method.	✓		

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
Sales Order Status	✓		
Sales Order Type	✓		