

Microsoft Dynamics SL

Microsoft Dynamics[®] SL

Microsoft Dynamics ISV Software Solution Test Guidelines



Date: March 30, 2009

Updated July 27, 2009

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

These test guidelines are for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

© 2009 Microsoft Corporation. All rights reserved.

Microsoft, Microsoft Dynamics, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Contents

- Introduction 6**
 - This Document 6
 - Program Overview..... 6
 - Contents of the Test Guidelines 6
 - Products Versions Supported..... 7
 - Types of Solutions..... 7
 - Expiration of the Test 8
 - More Information 8
- Testing Process 10**
- Documentation Requirements 11**
 - First-Time Software Test Requirements..... 11
 - Software Retest Requirements 12
- ISV Software Solution Requirements and Recommendations 13**
 - Development Requirements..... 13
 - 1.1 A .NET Framework–based ISV Application Must Be Compiled on .NET Framework 2.0 or Later and Must Pass the Required FxCop Tests 14
 - 1.2 Managed Assemblies Should Be Strong Named 15
 - 1.3 ActiveX Controls Should Be Digitally Signed..... 16
 - 1.4 An ISV Application Should Make Its Version Information Available 17
 - 1.5 An ISV Application That Has Its Own User Interface Must Have an About Window..... 18
 - User Assistance and Documentation Requirements..... 18
 - 2.1 An ISV Application Must Include User Documentation..... 18
 - 2.2 The ISV Must Provide Implementation Documentation..... 20
 - 2.3 The Documentation for an ISV Application Should Use the Microsoft Help Style Guidelines 21
 - User Experience and Usability Requirements..... 22
 - 3.1 An ISV Application Should Comply With User Interface Guidelines 22
 - 3.2 An ISV Application Must Not Restrict the Functionality of Microsoft Dynamics SL..... 23
 - Reporting Guidelines..... 24
 - 4.1 An ISV Application Should Follow the Microsoft Dynamics SL Reporting Guidelines 24
 - 4.2 An ISV Application Should Follow the SQL Server Reporting Services Implementation Guidelines 25
 - Trustworthy Computing Requirements..... 26
 - 5.1 An ISV Application Should Conform to the Microsoft Dynamics SL Security Model..... 26

Translation and Localization.....	27
6.1 An ISV Application Should Follow Globalization Rules	27
6.2 An ISV Application Should Separate Labels from Source Code	28
Technology Configurations and Platform Requirements.....	29
7.1 An ISV Application Must Support the Infrastructure that Microsoft Dynamics SL Supports.....	29
Installation and Setup Requirements	30
8.1 An ISV Application Must Include Documented Installation Procedures	30
8.2 The Installation Process for an ISV Application Must Correctly Register DLLs and COM Components	31
8.3 The ISV Must Document the Version and Service Pack of All Dependent Software Programs, Including Microsoft Dynamics SL, Required for Installing Its Application	32
8.4 The ISV Must Document the Microsoft Dynamics SL Modules Required for Using Its Application	33
8.5 The ISV Must Document the Procedures for Removing Its Application	33
8.6 An ISV Application Should Include Installable Demonstration Data.....	34
8.7 An ISV Application Must Not Adversely Affect Microsoft Dynamics SL After It Is Installed.....	35
Backup and Restore.....	36
9.1 The ISV Must Include Procedures to Back Up and Restore Its Application and Data	36
Extensibility and Customization Requirements	36
10.1 The ISV Should Document Its Application Extensibility and Customization Strategy	37
Upgrade and Maintenance	37
11.1 The ISV Must Provide Database Upgrade Scripts	38
11.2 The ISV Must Use File Versioning for DLLs and COM Components	38
Best Practice Guidelines	40
Trustworthy Computing Best Practices	40
BP1 The ISV Must Educate Staff on the Security Development Lifecycle Before Starting Design and Development Work	40
BP2 The ISV Must Establish and Follow Secure Development Best Practices	41
Design and Development Best Practices	43
BP3 The ISV Should Establish a Repeatable Process for Creating and Reviewing Design Specifications.....	43
BP4 The ISV Should Implement a Bug Tracking System	44
BP5 The ISV Should Remove Non-Functioning Code from the Application's Code Base	45

Appendix A: User Interface (UI) Guidelines	46
Application Windows	46
Edit Controls.....	46
Buttons	47
Other Controls and Toolbars.....	47
Icons and Symbols.....	48
User Assistance	48
Help	48
Screen Tips and Tool Tips	49
Messages.....	49
Appendix B: Microsoft Dynamics SL Consistency Verification Test.....	50
Identify the intended purpose of the product.	50
Example:.....	50
Describe the product purpose:	50
Identify primary functions.	50
List all primary functions:	51
Identify contributing functions.....	51
Example of a contributing function:	51
List all contributing functions:	51
Specify potential instabilities and challenging data.	51
Examples:	51
List all potential instabilities and challenging data:.....	51
Design and record a consistency verification test.	52
Prerequisites:.....	52
Required Information:	52
Test Procedure:	52

Introduction

This Document

The Microsoft Dynamics® SL ISV Software Solution Test Guidelines describes the technical requirements that an application must meet to integrate and operate with Microsoft Dynamics SL. For information about testing the requirements, see the "How to Comply" and "Test Methodology" sections in each requirement description.

Program Overview

Welcome to the ISV Software Solution Test Guidelines for Microsoft Dynamics SL. This document describes the requirements that an Independent Software Vendor (ISV) application must meet to integrate and operate with Microsoft Dynamics SL.

The goal of the test is to increase the quality of applications that run in the Microsoft Dynamics SL environment. The purpose of this test is to give the market the assurance that ISV applications built on Microsoft Dynamics meet technical requirements that help ensure a high standard. The test guidelines are designed to walk you through the test process and to help you ascertain that your application meets the requirements.

This document provides the following:

- An explanation of the testing process
- Definitions of the software requirements
- Descriptions of development, test, and documentation best practices

To pass the test, an ISV must demonstrate the development quality of its product and its ability as a software company to maintain and enhance that product in the future. Accordingly, there is a technical review and an in-lab inspection included in the test. The actual test is administered and conducted by a third-party vendor.

We welcome your comments and suggestions. Please contact dyncert@microsoft.com with your feedback.

Contents of the Test Guidelines

The Microsoft Dynamics SL ISV Software Solution Test Guidelines describe the test requirements, recommendations, and best practices. The guidelines are presented in individual subject-based modules, some of which may be common to other Microsoft Dynamics tests.

This document contains the following sections:

- Introduction (this section) explains the purpose and high-level requirements of the test.
- [Testing Process](#) describes how the testing process works from qualification through communication of test results.
- [Documentation Requirements](#) provides a summary of the documentation that you must submit with your application.
- [Test Requirements](#) defines in detail each requirement category, how these requirements are tested, and what you can do to make sure that your application meets the requirements.

Products Versions Supported

The intent of the test is to support Microsoft's latest shipping version of a product. For that reason, ISV applications submitted for testing must run on Microsoft Dynamics SL release 7.0 or later with the latest service pack installed.

Types of Solutions

Microsoft Dynamics solutions fall into three general categories and three setup complexity levels. The category and setup complexity of a solution determines (in part) the type and complexity of the testing requirements and the costs associated with testing the solution.

Figure 1 shows the different solution categories and setup complexity levels.

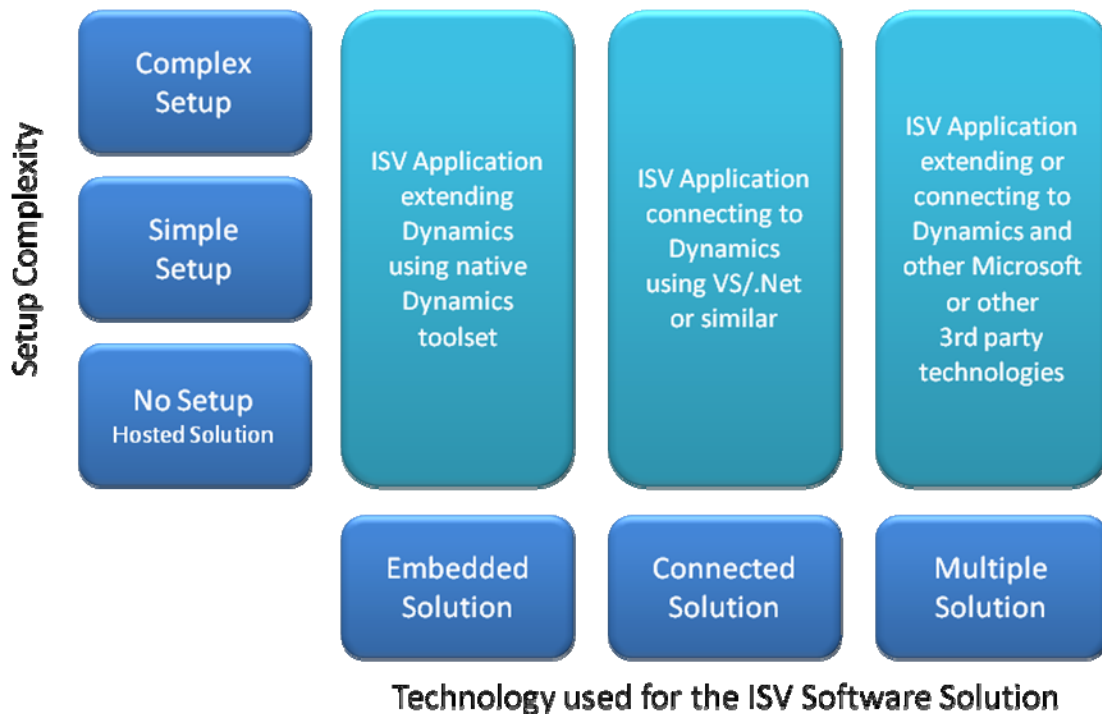


Figure 1
Solution categories and setup complexity levels

ISV software solution complexity falls into one of the following categories (listed from least complex to most complex):

- An *embedded* solution is an ISV application that extends Microsoft Dynamics using only the tools provided with the Microsoft Dynamics product. For example, embedded solutions are built in the applicable native development environment, including the Microsoft Dynamics SL Software Development Kit (SDK, formerly VB Tools) environment, Microsoft Dynamics GP Dexterity environment, Microsoft Dynamics NAV C/SIDE, C/AL environment, or the Microsoft Dynamics AX Morph, X++ environment.
- A *connected* solution is an ISV application that uses Microsoft Visual Studio, the .NET Framework, or similar tools to connect to the Microsoft Dynamics product.

Typically, a connected solution refers to a standalone product that interoperates with the core Microsoft Dynamics product by using it as a business rules engine. The solution might establish interoperability by using Web Services, .NET Framework assemblies, COM interop, or some other means. The solution might or might not be .NET Framework–based; however, it must run on a Microsoft operating system.

- A *multiple* solution is one that extends or connects to Microsoft Dynamics and other Microsoft or other third-party technologies.

Setup complexity falls into one of the following categories (listed from least complex to most complex):

- No setup (potentially a *hosted* solution) provides services to customers, who do not have to purchase, install, or maintain the software or hardware. Hosted solutions have no setup requirements for end users; however, installing and configuring a hosted solution can be extremely complex and the test vendor may not have the hardware, custom software, or services that the solution requires.
- A *simple* setup is one that the test vendor can install and configure without requiring a restorable backup, virtual PC (VPC), or other additional assistance.
- A *complex* setup is one that the test vendor cannot completely replicate; for example, solutions that require specific hardware, custom software, or back-end services that the vendor cannot duplicate.

Expiration of the Test

Test results are valid for 24 months. When retested, the ISV application must be updated to support the latest Microsoft Dynamics release including the latest service pack.

More Information

For more information about the functionality of Microsoft Dynamics SL, see the Microsoft Dynamics SL Home Page at <http://www.microsoft.com/dynamics/SL/default.aspx>.

For more information about the Microsoft Partner Program, see the Microsoft Worldwide Partner Portal Home Page at <http://partner.microsoft.com>.

For more information about how the ISV test helps you earn partner program points, see the ISV Software Testing Framework page at <http://partner.microsoft.com/global/program/competencies/40011374>.

For more information about the Microsoft Dynamics ISV/Software Solutions competency, see the Microsoft Dynamics Testing for ISVs page at <http://partner.microsoft.com/global/program/competencies/40013116>.

For more information about developing applications that integrate with Microsoft Dynamics SL, see the following user manuals, which are included with the software:

- *Microsoft Dynamics SL Software Development Kit (SDK)* (SL_SDK.pdf) documents the same development platform used by Microsoft developers to create Microsoft Dynamics SL. External developers can use it for the following purposes:
 - Create tightly-integrated applications with the same look and feel as the rest of Microsoft Dynamics SL;
 - Provide support for the Customization Manager module;
 - Provide integration with security functionality;

- Provide built-in multi-user contention checking;
- Accelerate speed of development.
- *Customization Manager Guide* (SL_CU.pdf) describes how users, IT staff, and consultants can modify standard Microsoft Dynamics SL screens quickly and easily. Modification capabilities range from simple changes that the system administrator or user can make to complex customizations that are best performed by a programmer or consultant with experience in Visual Basic programming and SQL syntax. Capabilities of the Customization Manager module include implementing customizations that apply to a single user or to all users.
- *Customization Manager Reference – Visual Basic for Applications* (SL_VB.pdf) contains Visual Basic for Applications (VBA) coding guidelines and a complete list of Microsoft Dynamics SL API references.
- *Object Model Reference Guide* (SL_OB.pdf) documents the object model for Visual Basic programmers, including VBA programmers who are working in the standard scripting environments of Microsoft Office applications and are looking to automate Microsoft Dynamics SL through Visual Basic client applications that they will write.
- *Visual Basic® 2005 Conversion Toolkit for Microsoft Dynamics™ SL* (SL_Conversion.pdf) provides instruction and guidance for application developers using the Conversion Toolkit, which is designed to translate Microsoft Dynamics SL applications that were developed using Microsoft Visual Basic 6.0 for use in a Microsoft Visual Basic .NET environment.
- *Microsoft Dynamics SL Reporting Guide* (SL_ROI.pdf) describes the reporting options available in Microsoft Dynamics SL.

These manuals are available on the Microsoft Dynamics SL product DVD and can also be downloaded from the [PartnerSource Web site](#). This secured partner portal also contains sample applications developed using Customization Manager with VBA (Visual Basic for Applications). Microsoft Certified Partners who are certified on Microsoft Dynamics SL can log on to PartnerSource and download the [Sample Applications for Customization Manager with VBA](#).

Testing Process

Microsoft offers ISV software solution testing through an independent third-party test vendor. ISVs register for the test by visiting the test vendor's Web site referenced on the Microsoft Web page. The vendor site contains a description of the test as well as an application form with a published test fee schedule.

Depending on the type of solution (embedded, connected, or multiple) and the solution setup (simple or complex), different test methods will apply, for which the test fee will vary. ISVs can make their products available to the test vendor for testing by using any of the following methods:

- Providing the software with installation instructions for the test vendor to install
- Sending a virtual server image of a working configuration of the product to the test vendor
- Using an interactive Live Meeting session to provide access to a working configuration of the product

After you register your software solution and pay the test fee, the test vendor will contact you with detailed information about the testing process you have selected. For processes involving shipping software or virtual server images to the test vendor, you can choose to send the software on media (CD/DVD), upload to an FTP server, or have the test vendor download from your server. If you choose to use Live Meeting to provide access to your solution, the test vendor will contact you to schedule the session.

The following requirements are particularly important:

- Pre-qualification is required. You are responsible for making certain that your solution and organization meet the requirements for submitting and maintaining a Microsoft Dynamics-based solution.
- You must submit a number of documents as part of the test. These are identified in the appropriate test modules, as well as in a summary checklist. See the [Documentation Requirements](#) section of this guide.
- You must upload documents and your solution to the test vendor's servers for testing. If your setup is complex, you must be prepared to use Microsoft Live Meeting to demonstrate the solution to the test vendor.

Documentation Requirements

The checklists in this section describe the items that you must include when you submit your software solution for testing or retesting.

First-Time Software Test Requirements

The following checklist describes the items that you must include when you submit your application for first-time testing. Note that a single document might contain information that meets multiple requirements. For that reason, the actual number of documents might be significantly smaller than the number of items on this list. Some of the documentation requirements apply only in some situations. Refer to the full description of the requirement for more information.

Requirement	Check
The ISV solution (your application software and associated documentation), which can be in the form of a CD or other distributable media. You can also use a virtual server image or Live Meeting session to demonstrate and provide access to your software.	
A description of the business functionality that the solution provides and examples of key usage scenarios (see Appendix B, Consistency Verification Test).	
Explanation and justification for any FxCop errors, if needed. See Requirement 1.1 .	
User documentation to assist customers in learning how to use your solution. See Requirement 2.1 .	
Implementation documentation appropriate for Microsoft Certified Partners or others who intend to deploy your solution. See Requirement 2.2 .	
Description of the security model used by your application. Include a copy of the license or other unlocking mechanism required for application registration, if needed. See Recommendation 5.1 .	
Description of the localization options supported by your application. See Recommendation 6.2 .	
List of vendor or third party controls. See Requirement 8.1 .	
Description of all registry settings generated during installation. See Requirement 8.2 .	
List of all components that the ISV application uses and required version information. This includes external components. See Requirement 8.3 and Requirement 8.4 .	
List of all resources that the ISV application adds to the Microsoft Dynamics SL application and complete instructions for uninstalling the ISV application. If certain aspects of application installation, such as stored procedures, cannot be removed, you must state this in the documentation. See Requirement 8.5 .	
Sample data for testing – this may or may not be part of the core application installation. See Recommendation 8.6 .	
Description of backup and restore procedures. See Requirement 9.1 .	
Customization and extensibility documentation that explains how to extend your application. See Recommendation 10.1 .	

Scripts to update databases from a previous version, if needed. See Requirement 11.1 .	
A training log that shows that ISV staff have met the security education requirements for trustworthy computing. See Best Practice BP1 .	
Design specification for the application. See Best Practice BP3 .	
Sample bug report. See Best Practice BP4 .	
Description of and justification for any exceptions to best practices guidelines. Prepare a status report to show the progress you are making towards achieving compliance with the guidelines.	

Software Retest Requirements

The following checklist describes the documentation that you must include with your solution when you submit it for update testing. Note that a single document might contain information that meets multiple requirements. For that reason, the actual number of documents might be significantly smaller than the number of items in this list.

Requirement	Check
The new release of the ISV solution (your application software) and product documentation. This can be in the form of a CD or other distributable media, or you can use a virtual server image or Live Meeting session to demonstrate your software.	
A <i>What's New</i> document that describes the business functionality of the new release.	
A description of changes to objects and components.	
A description of changes to the data model.	
Upgrade scripts for the new release.	

ISV Software Solution Requirements and Recommendations

The ISV software solution requirements ensure that ISV applications integrate with Microsoft Dynamics SL without causing system problems or errors. Microsoft and third party test vendors worked together to define the minimum requirements that an ISV application must meet to operate successfully in a Microsoft Dynamics SL environment.

Note: The test does not validate the marketability or relevance of ISV application functionality.

This section describes the test requirements and recommendations. It also describes the procedures for verifying that each requirement is met. In this document, the word “must” in the text of a requirement means that the item or feature is not optional. The word “should” means that the item or feature is recommended and its inclusion is a best practice but is not strictly required. However, these recommendations will be considered for inclusion as requirements in later versions of this test.

Some requirements are technology-specific and do not apply to all ISV solutions. Therefore, each requirement indicates the type of ISV technology to which it applies. Additionally, an ISV solution might include several technologies. In these situations, the vendor will test those parts of the solution that use the technologies to which the requirement applies.

If a requirement is defined as applicable to an *SDK-based* solution, it applies to either of the following:

- Any code (either business logic or code that implements an integration to an external component) written using the Microsoft Dynamics SL SDK (or converted from VB Tools using the Visual Basic® 2005 Conversion Toolkit for Microsoft Dynamics™ SL) if the vendor in-lab test is performed directly on the code
- Any solution that includes SDK code if an in-lab test is not performed directly on the code

Similarly, if a test is defined as applicable to *External*, it applies to either of the following:

- Any code not written using the SDK (including DLLs, ActiveX controls, services, applications that have their own user interface, and so on) if the requirement is code-specific
- Any solution that includes such code if the requirement is not code-specific

Each requirement includes a table that indicates the type of technology (SDK-based or external) and type of solution setup (simple, complex, and host-based) to which the requirement applies.

Development Requirements

Your application must meet the following development requirements:

- [Requirement 1.1: A managed code ISV application must be compiled on .NET Framework 2.0 or later and must pass the required FxCop tests](#)
- [Requirement 1.2: Managed assemblies must be strong named](#)
- [Recommendation 1.3: ActiveX controls should be digitally signed](#)
- [Recommendation 1.4: An ISV application should make its version information available](#)
- [Requirement 1.5: An ISV application that has its own user interface must have an About window](#)

1.1 A .NET Framework–based ISV Application Must Be Compiled on .NET Framework 2.0 or Later and Must Pass the Required FxCop Tests

Type	Test Method	Technology	Solution Category		
Required	In-lab test	External	Simple	Complex	Hosted
		Managed code only	✓	✓	✓

Summary and Intent

.NET Framework–based ISV applications must be compiled on .NET Framework 2.0 or later and should use the current version of Microsoft .NET Framework. In addition, the application must pass the required FxCop tests. FxCop 1.36 is a code analysis tool that checks .NET managed code assemblies for conformance to the Microsoft .NET Framework design guidelines.

Note: FxCop works only with managed code.

An ISV can download FxCop from the FxCop Web site. FxCop uses reflection, MSIL parsing, and call graph analysis to inspect assemblies for 200 possible defects.

Resources

For more information, refer to the following:

- .NET Framework Web site: <http://go.microsoft.com/fwlink/?linkid=147530>
- FxCop Web site: <http://go.microsoft.com/fwlink/?linkid=140144>
- FxCop page in the .NET Framework Developer Center: <http://msdn.microsoft.com/en-us/library/bb429476.aspx>

How to Comply

Download FxCop from the FxCop Web site. FxCop includes the following rule libraries based on the .NET Framework design guidelines that are loaded by default when a new project is created:

- **COM** – Rules that detect COM Interop issues.
- **Design** – Rules that detect potential design flaws. These coding errors typically do not affect the execution of your code.
- **Globalization** – Rules that detect missing or incorrect usage of information related to globalization and localization.
- **Naming** – Rules that detect incorrect casing, cross language keyword collisions, and other issues related to the names of types, members, parameters, namespaces, and assemblies.
- **Performance** – Rules that detect elements in your assemblies that will degrade performance.
- **Security** – Rules that detect programming elements that leave your assemblies vulnerable to malicious users or code.
- **Usage** – Rules that detect potential flaws in your assemblies that can affect code execution.

Rules are assigned one of five importance levels:

- **Critical error** – Issues that are highly visible, that prevent code from operating correctly in common scenarios, or both. You should resolve critical error messages first. You should exclude these only after carefully assessing the impact of ignoring the error.
- **Error** – Issues that have less impact on usability and behavior than critical errors. You should not exclude these errors without careful analysis.
- **Critical warning** – Issues that typically have little or no negative impact on code behavior. These messages may indicate issues with code maintainability and less-than-optimal choices for visible elements. However, these messages are considered errors and you should review them closely before you exclude them.
- **Warning** – Issues that are typically concerned with doing things correctly to keep your code base stable, extensible, and maintainable.
- **Informational** – Messages returned by rules that report information about a system, rather than detecting errors in a system.

To pass the requirement, the ISV application must act on all “critical errors” of all issue types, as well as on all security issues with importance level “error.” However, it is a good practice to act on all issue types of all importance levels.

No errors should be reported in any rules in the security category. Any security exceptions must be reviewed by Microsoft. Exceptions can be granted for warnings. For example, you could suppress an error by explaining the reason for violating the rule and why this is not a problem. It is possible to suppress a violation either in source or in an FxCop project file. See [http://msdn2.microsoft.com/en-us/library/ms244717\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/ms244717(VS.80).aspx) for in-source suppressions. If suppressions are done in an FxCop project file instead, the ISV should also provide this project file to the test vendor.

Test Methodology

The test vendor will use FxCop to analyze the ISV application (typically the binaries). If FxCop reports any critical security errors, the ISV must provide a written explanation and justification (in the tool or in a separate document). No errors should be reported in any rules in the security category (exceptions can be granted for warnings). All security exceptions must be reviewed by Microsoft.

Criteria for Passing

This requirement is mandatory. If the ISV solution does not pass this requirement, it will fail the test.

1.2 Managed Assemblies Should Be Strong Named

Type	Test Method	Technology	Solution Category		
Recommended	In-lab review	External	Simple	Complex	Hosted
		Managed Code	✓	✓	✓

Summary and Intent

The ISV should apply strong naming to managed assemblies for security purposes. The Sn.exe tool provided with the Microsoft Visual Studio® 2005 development system supports the proper use of strong names.

Resources

The Sn.exe tool provided with the Microsoft Visual Studio® .NET development system supports the proper use of strong names. For more information, refer to <http://go.microsoft.com/fwlink/?linkid=142166>.

How to Comply

The ISV should apply strong naming to managed assemblies. The exception is if the ISV application uses a vendor or third party assembly, the assembly does not need to be signed. The ISV should provide a list of vendor or third party assemblies.

Test Methodology

The test vendor will use the Sn.exe tool provided with Visual Studio .NET to verify the proper use of strong names.

Criteria for Passing

This is a recommendation only. Failure to comply with this recommendation will not cause the application to automatically fail the test.

1.3 ActiveX Controls Should Be Digitally Signed

Type	Test Method	Technology	Solution Category		
Recommended	In-lab review	External	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

Digital signing helps users decide whether to trust a control and assures users that files have not been tampered with. The user should not be placed in the position of having to accept unsigned controls.

Resources

Code signing certificates are available from several vendors, as described at <http://go.microsoft.com/fwlink/?linkid=57475>.

Information about using the Microsoft Windows SDK Sign Tool is available on MSDN at: [http://msdn.microsoft.com/en-us/library/8s9b9yaz\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/8s9b9yaz(VS.80).aspx).

How to Comply

After the ISV obtains a code-signing certificate, the ISV should use the Microsoft Windows SDK Sign Tool to sign the files. The exception is that if the ISV application uses a vendor or third-party assembly or ActiveX control, the control does not need to be signed. The ISV should provide a list of vendor or third party controls.

Test Methodology

During testing, the test vendor will note any warnings about ActiveX controls without valid certificates.

Criteria for Passing

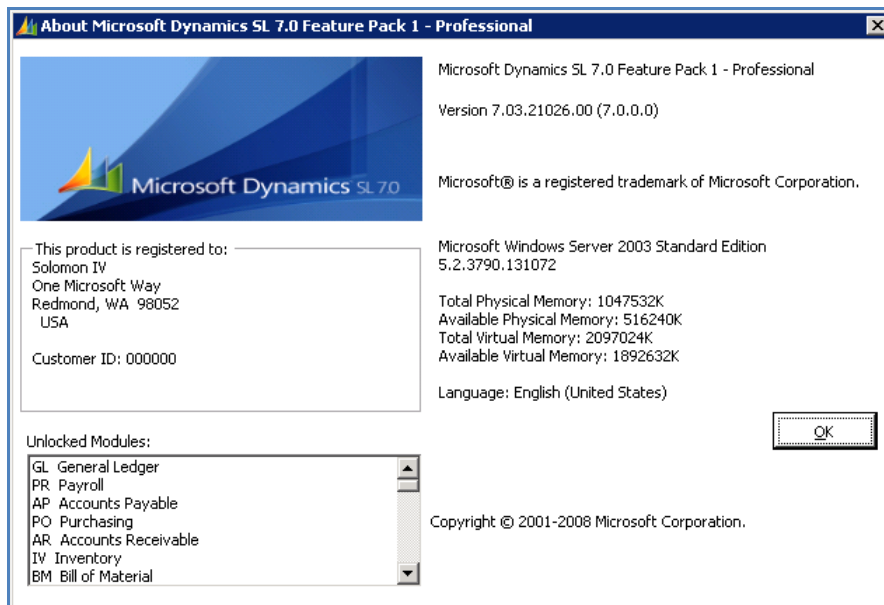
This is a recommendation only. Failure to comply with this recommendation will not cause the application to automatically fail the test.

1.4 An ISV Application Should Make Its Version Information Available

Type	Test Method	Technology	Solution Category		
Recommended	In-lab test	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

For support purposes, a user should be possible to identify the version of the ISV application from the UI. This information might, for example, be included in an About box such as the one displayed below.



Resources

For more information, refer to the *Microsoft Dynamics SL Software Development Kit (SL_SDK.PDF)*, located on the Microsoft Dynamics SL product DVD and optionally installed in the User Guides folder of the Microsoft Dynamics SL installation. Use the AboutBox method, described in **Reference | Methods | AboutBox Method**.

How to Comply

Follow the guidelines in the *Microsoft Dynamics SL Software Development Kit*.

Test Methodology

The test vendor will verify there is an About window accessible on the Help menu and that the About window contains version information.

Criteria for Passing

This is a recommendation only. Failure to comply with this recommendation will not cause the application to automatically fail the test.

1.5 An ISV Application That Has Its Own User Interface Must Have an About Window

Type	Test Method	Technology	Solution Category		
Required	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

An ISV application that has its own user interface must have an About window that is accessible from the Help menu. The About window must display the name of the application, contact details for the ISV, and version and build information for the application.

Resources

None

How to Comply

Make sure that version information for your application is available in the UI About window.

Test Methodology

The test vendor will verify there is an About window that is accessible on the Help menu.

Criteria for Passing

This requirement is mandatory. If the external ISV application does not include an About window, it will fail the test.

User Assistance and Documentation Requirements

Your application must comply with the following user assistance and documentation requirements:

- [2.1 An ISV Application Must Include User Documentation](#)
- [2.2 The ISV Must Provide Implementation Assistance](#)
- [2.3 The ISV Solution Online Documentation Should Use the Microsoft Help Style Guidelines](#)

2.1 An ISV Application Must Include User Documentation

Type	Test Method	Technology	Solution Category		
Required	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

Your application must have some form of documentation that is easy for the user to access and navigate. This can be in the form of a help file (with optional field-level context sensitivity) or access to an electronic user guide. Additional user assistance can be provided in the form of UI text (labels or captions) and tooltips for UI fields and buttons. However, UI text cannot be substituted for online help.

Note: Recent security changes made to the Windows Server 2003 operating system now block the ability to view remote content. The workarounds provided for system administrators involve modifying the Windows registry on the server or modifying group policy settings. Because of these security changes, help files should be installed to users' workstations (the restrictions do not apply to content files that are installed locally). If your application includes client components, include your help files with the client components. For more information, refer to [Microsoft Knowledge Base article 902225](#).

Documentation for an embedded in-product Microsoft Dynamics SL application should provide a user experience that is consistent with the base documentation provided with Microsoft Dynamics SL. Therefore, your online documentation should be in the compiled HTML help (.CHM) file format.

Note: Although the compiled HTML help that is standard with Microsoft Dynamics SL does not support adding books to the help table of contents, a book can be added to users' menus or to favorites that appear on the Microsoft Dynamics SL window in their home pane.

In an embedded in-product solution, users must be able to see help topics by pressing the F1 function key, which starts Microsoft Dynamics SL help. Additional information may potentially be available in electronic user guides ("soft copy") or printed manuals ("hard copy," which is not recommended because printed manuals can be accessed by only one user at a time). Linking a help file to an application screen is described in the "Adding Custom HTML Help" section of the Microsoft Dynamics SL Software Development Kit (SDK).

If your solution is an external application, it should also provide some form of online help. In addition, users should be able to see help topics by pressing the F1 function key. If online help is not provided with the application, the electronic user guides must be installed automatically with the application.

Help files and electronic user guides must include the overall purpose of each screen or report, when and why the screen should be used, and how its use will affect the rest of the system (downstream processing implications).

Note: This requirement does not apply to user interfaces designed for special devices, such as hand-held devices, cash registers, and so on.

Resources

For more information, refer to following:

- MSDN HTML Help: [http://msdn2.microsoft.com/en-us/library/aa189109\(office.10\).aspx](http://msdn2.microsoft.com/en-us/library/aa189109(office.10).aspx)
- "Adding Custom HTML Help" in the *Microsoft Dynamics SL Software Development Kit (SL_SDK.PDF)*, located on the Microsoft Dynamics SL product DVD and optionally installed in the User Guides folder of the Microsoft Dynamics SL installation
- MSDN HelpProvider Component (Windows Forms): [http://msdn2.microsoft.com/en-us/library/2ksk25ts\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/2ksk25ts(VS.80).aspx)

How to Comply

Follow the guidelines in the documents listed in the Resources section above.

Test Methodology

The test vendor will review your help documentation for compliance and usability. The test vendor will review a representative sample of application modules to make sure that help is available when a user presses F1 and that the help content is appropriate for the current application window.

Criteria for Passing

This requirement is mandatory. If the ISV solution does not provide some form of online documentation, it will fail the test.

2.2 The ISV Must Provide Implementation Documentation

Type	Test Method	Technology	Solution Category		
Required	In-lab review	All	Simple	Complex	Hosted
			✓	✓	

Summary and Intent

You must include implementation assistance in your documentation. The documentation should reference the system requirements, explain the configuration of setup options (including clear explanations of the implications of each choice in downstream processing), and describe guidelines for secure deployment. If your solution has a dependency on an existing Microsoft Dynamics SL module or feature being configured in a specific manner, this constraint must be presented clearly.

Implementation assistance can be in the form of a help file or an online document. If possible, implementation information should be included in your solution, but distribution via download from your Web site is acceptable for a connected solution that runs on a device separate from the server or workstation, such as a handheld device. If you do not use partners to sell your solution, you must provide implementation information to customers as well. If you allow your product to be implemented only by your own employees, an internal document explaining how to implement your product must be provided to the test vendor.

ISV partners and customers who use or deploy an application must be able to successfully deploy, configure, and manage the application in an existing Microsoft Dynamics SL environment. Your documentation must provide information that allows your partners and customers to successfully install or upgrade your solution in such an environment.

Resources

The following link opens a page containing documents published for Microsoft Dynamics SL Release 7.0: https://mbs.microsoft.com/customersource/documentation/userguides/UserGuides_SL70. Included are links to the System Manager Guide, Security Guidelines, System Requirements Guide, and the user guides for all modules.

- If your solution includes a setup screen, used for configuring options that affect how the software behaves for all users, refer to the section “Setting up Labor Capture” in the Time and Expense for Projects user guide. The section describes configuring the options for time entry using the *Time and Expense Setup* screen.

- If your solution uses multiple screens for configuring setup options, refer to the section “Setting up the General Ledger” in the General Ledger user guide. The section describes setting up accounts, subaccounts, ledgers, and allocation groups using a variety of screens.

You can use these documents as samples for your documentation.

It is recommended that you also make your documentation available for download from your Web site. Doing so enables you to issue documentation updates when necessary.

How to Comply

To meet this requirement, you must document adequate system requirements and installation, configuration, and upgrade procedures for implementing your application in a new and in an existing Microsoft Dynamics SL environment.

Test Methodology

The test vendor will review your documentation to verify that you have included adequate implementation information, which should contain the following sections:

- Description of the solution (the problem it solves)
- Hardware environment
- Operating system requirements
- Installation checklist
- Installation procedures
- Secure deployment guidelines
- Operational checklist (daily, monthly, and annual procedures, as well as how to perform backups and so on)

Note: The test vendor will validate that the required information is included. The test vendor will not test the procedures or verify the quality of the information.

In addition, if your application contains a setup screen, the test vendor will configure the options using all available selections and verify that the effect of each selection is as described in the documentation.

Criteria for Passing

This requirement is mandatory. If the ISV solution documentation does not include implementation assistance, the solution will fail the test.

2.3 The Documentation for an ISV Application Should Use the Microsoft Help Style Guidelines

Type	Test Method	Technology	Solution Category		
Recommended	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

The online documentation for your application should follow Microsoft Help style guidelines. It should tell the user how to perform specific tasks and it should be easy for the user to understand. Documentation for an ISV application should provide a user experience that is consistent, both in terms of writing style and depth of information, with the base documentation provided with Microsoft Dynamics SL.

Resources

For more information about style guidelines, see the MSDN online Help style guidelines at <http://msdn2.microsoft.com/en-us/library/ms997600.aspx>. For information about linking a help file with an application, refer to 2.1 An ISV Application Must Include User Documentation.

How to Comply

Make sure that you have online help and that it provides meaningful information. The guidelines on MSDN will help you to create appropriate content.

Test Methodology

The test vendor will review your Help documentation for style, accuracy, and usability. The vendor will review a representative sample of application modules to make sure that Help topics are appropriate, easy to understand, correct, and adhere to style and user interface guidelines.

Criteria for Passing

This is a recommendation only. Failure to comply with this recommendation will not cause the application to automatically fail the test.

User Experience and Usability Requirements

Your application must comply with the following user experience and usability requirements:

- [3.1 An ISV Application Should Comply With User Interface Guidelines](#)
- [3.2 An ISV Application Must Not Restrict the Functionality of Microsoft Dynamics SL](#)

3.1 An ISV Application Should Comply With User Interface Guidelines

Type	Test Method	Technology	Solution Category		
Recommended	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

Users of your application should have a user experience consistent with Microsoft Dynamics SL or with the Microsoft guidelines. Therefore, the UI for your application should conform to Microsoft Dynamics SL UI guidelines appended to this document or to Windows guidelines, whichever is appropriate for your solution.

An embedded ISV application should follow the UI guidelines and naming conventions described in the *Microsoft Dynamics SL Software Development Kit (SDK)* (SL_SDK.pdf). If you use established guidelines and naming conventions, other developers can work with your application more easily.

At a minimum, the application must display appropriate messages for common user errors such as leaving a required field blank and entering an invalid value in a validated field. For messages unique to the ISV application, each message must include constructive information to assist the user in resolving the error.

You should inform the test vendor of the guidelines used in your application. If you used a Microsoft guideline from MSDN, provide the test vendor with the link to the appropriate guideline. If you used the guidelines in the *Microsoft Dynamics SL Software Development Kit (SDK)* (SL_SDK.pdf) or the appendix, inform the test vendor of this.

Resources

- [Windows User Experience Interaction Guidelines](#)
- [Design Guidelines for Windows Mobile 6](#)

See [Appendix A](#) for some basic UI guidelines. ISV applications serve a variety of purposes and support different technologies (Windows, Web, and so on) and devices (SmartPhones, RFID scanners, and so on). Because of the various ways that solutions can be developed to support and extend Microsoft Dynamics SL, we recommend going to MSDN and searching for “user interface” for the platform or technology appropriate for the solution.

How to Comply

To produce a compliant Microsoft Dynamics SL application, follow these general recommendations:

- Read and understand the user interface guidelines and checklists in the appendix.
- Become familiar with the Windows User Experience guidelines on MSDN.
- For .NET Framework–based code, you should identify and adhere to a standard set of UI Guidelines.

Test Methodology

The test vendor will open a random selection of windows added by the ISV solution and confirm that each one follows the appropriate set of UI guidelines according to the guidelines you specify to the vendor. In addition, while performing data entry in the ISV application, the test vendor will skip a field that the user documentation states is required and verify that an appropriate message appears. The vendor will also enter an invalid value in a field described in user documentation as a validated field and verify that an appropriate message appears.

Criteria for Passing

This is a recommendation only. Failure to comply with this recommendation will not cause the application to automatically fail the test.

3.2 An ISV Application Must Not Restrict the Functionality of Microsoft Dynamics SL

Type	Test Method	Technology	Solution Category		
Required	In-lab review	All	Simple	Complex	Hosted



Summary and Intent

An ISV application must leave the native functionality of Microsoft Dynamics SL intact. When the ISV application is added to or removed from a Microsoft Dynamics SL system, the ISV application must not cause any damage (whether intentional or unintentional) or restrict the normal functioning of Microsoft Dynamics SL.

Resources

None

How to Comply

Read and understand the Microsoft Dynamics SL SDK to gain an understanding of how native Microsoft Dynamics SL functionality operates so that your application does not inadvertently interfere with the normal functioning of Microsoft Dynamics SL.

Test Methodology

After installing the ISV application, the test vendor will open a random selection of Microsoft Dynamics SL screens and confirm that they open without error. The test vendor will perform the same verification after removing the ISV application.

Criteria for Passing

This requirement is mandatory. If after installing the ISV application, a native Microsoft Dynamics SL screen fails to open or displays an error message, the application will fail the test.

Reporting Guidelines

Your solution should comply with the following reporting guidelines:

- [4.1 An ISV Application Should Follow the Microsoft Dynamics SL Reporting Guidelines](#)
- [4.2 An ISV Application Should Follow the SQL Server Reporting Services Implementation Guidelines](#)

4.1 An ISV Application Should Follow the Microsoft Dynamics SL Reporting Guidelines

Type	Test Method	Technology	Solution Category		
Recommended	None	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

ISV applications should provide reporting that is consistent with the reporting functions of Microsoft Dynamics SL.

Resources

For more information, refer to the *Microsoft Dynamics SL Reporting Guide (SL_ROI.PDF)*, available on the Microsoft Dynamics SL product DVD. Also, refer to existing reports in Microsoft Dynamics SL.

How to Comply

You should provide a list of reports as part of your application submission. If your application is .NET Framework–based, provide a description of the reporting guidelines you used.

Test Methodology

The test vendor will review your application to verify that it meets Microsoft Dynamics SL reporting guidelines. The test vendor will randomly test up to four reports on the list to verify that they are installed and that the user experience is consistent with other reports delivered in the core product.

Criteria for Passing

This is a recommendation only. Failure to comply with this recommendation will not cause the application to automatically fail the test.

4.2 An ISV Application Should Follow the SQL Server Reporting Services Implementation Guidelines

Type	Test Method	Technology	Solution Category		
Recommended	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

Microsoft Dynamics SL provides implementation guidelines for reports using SQL Server Reporting Services (SSRS). These guidelines help ensure that SSRS-based reports are surfaced in a consistent manner and function accurately in Microsoft Dynamics SL.

Resources

For more information, refer to the *Microsoft Dynamics SL Reporting Guide (SL_ROI.PDF)*, available on the Microsoft Dynamics SL product DVD. Also, refer to existing reports in Microsoft Dynamics SL.

How to Comply

Use the SSRS reporting implementation guidelines when you design and implement reports. Include design documentation that describes how your report complies with the guidelines.

Test Methodology

The test vendor will review your application to verify that it meets SSRS reporting implementation requirements.

Criteria for Passing

This is a recommendation only. Failure to comply with this recommendation will not cause the application to automatically fail the test.

Trustworthy Computing Requirements

Your application must comply with the following trustworthy computing requirements:

- [5.1 Application Should Conform to the Microsoft Dynamics SL Security Model](#)

5.1 An ISV Application Should Conform to the Microsoft Dynamics SL Security Model

Type	Test Method	Technology	Solution Category		
Recommendation	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

Your application should use the Microsoft Dynamics SL security model. This recommendation is intended to prevent an ISV application from reducing the security level in the standard Microsoft Dynamics SL product by ensuring that application data is not available to unauthorized users. Embedded application screens (including tabs and sub-screens) and reports accessed from the Microsoft Dynamics SL application (parent) window must follow the Microsoft Dynamics SL Access Rights methodology for securing access.

Resources

For more information, refer to the *Microsoft Dynamics SL Software Development Kit (SL_SDK.PDF)*, located on the Microsoft Dynamics SL product DVD and optionally installed in the User Guides folder of the Microsoft Dynamics SL installation.

How to Comply

If parts of your ISV application do not use the Microsoft Dynamics SL security model, you should document how security is enabled for that part of your application. You should explain how to set up users securely and how to assure that standard security cannot be compromised. You can do this in a separate security-hardening guide or as a section in the implementation guide.

Bypassing standard security can be a problem if data is changed while the application is executing under a general user identity (rather than using a specific user identity) or when a specific user imports batches of data using the Transaction Import utility. In these cases, you should create a threat model to determine if there is a problem and you must document how to mitigate the risk.

Even if you write only SDK code, you should still be aware of the security implications while importing/exporting data. You should particularly be careful when communicating with other servers.

Your ISV application should include documentation that describes how security should be enabled on the additional forms and reports added to Microsoft Dynamics SL. The documentation should include suggestions for roles that should be created and how to set permissions for them. Your documentation should supplement the [Microsoft Dynamics SL Security Guidelines](#) (MicrosoftDynamicsSLSecurity.pdf), which is also located on the Microsoft Dynamics SL product DVD and optionally installed in the User Guides folder of the Microsoft Dynamics SL installation.

Test Methodology

The test vendor will verify that your application meets the recommendations as follows:

- Tools and processes: The vendor will perform a qualitative review to make sure that required processes and tools are in place and that processes were successfully executed. (The vendor will not audit the output of your tools and processes.)
- FxCop: The vendor will run FxCop and will require that you fix all reported issues.
- .NET Framework SDL Shared Section requirements (no writable PE shared sections): The vendor will run tools and require that you fix all reported issues.

Criteria for Passing

This requirement is mandatory. If the ISV solution does not follow security guidelines, it will fail the test.

Translation and Localization

Your application should comply with the following best practices:

- [6.1 An ISV Application Should Follow Globalization Rules](#)
- [6.2 An ISV Application Should Separate Labels from Source Code](#)

6.1 An ISV Application Should Follow Globalization Rules

Type	Test Method	Technology	Solution Category		
Required	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

The following recommendations are important for the successful translation and localization of a Microsoft Dynamics SL certified ISV solution.

The .NET Framework provides built in support for globalization.

Resources

For more information, see the following:

- [Microsoft Global Development and Computing Portal](#)
- [MSDN Developer Center: Visual Studio Globalization](#)

How to Comply

You should ensure that the application can be set up using local settings such as date, time, currency, and so on.

Test Methodology

During the in-lab test, the test vendor will perform a qualitative review to determine whether your application follows globalization rules with respect to labels and data types for time and dates.

For .NET Framework–based applications, the FxCop tool is used to check for compliance.

Criteria for Passing

This is a recommendation only. Failure to comply with this recommendation will not cause the application to automatically fail the test.

6.2 An ISV Application Should Separate Labels from Source Code

Type	Test Method	Technology	Solution Category		
Recommended	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

The intent of this recommendation is to help ISVs build globalized software. That is, ISV solutions should be capable of being localized for the same markets that Microsoft Dynamics SL serves. Therefore, your application should follow localization best practices.

If you plan to localize your application into other languages, you should use specialized tools that recycle translations of repeated text and resize application UI elements to allow for localized text and graphics. In addition, all label files should be translated into the languages where the ISV application is sold.

Even if you do not plan to localize your application into other languages, you still should consider how your application will operate with other language configurations. For example, if you use U.S. English regional settings, your application might not run smoothly on a computer with different regional settings.

Resources

For more information, refer to the following:

- [Microsoft Global Development and Computing Portal](#)
- [MSDN Developer Center: Visual Studio Globalization](#)
- [MSDN Library: Localization Planning](#)

How to Comply

You should provide an overview of how you plan to localize your application. You should also provide evidence that strings are separated from code.

If your application supports only one language, you should document this clearly (in your sales/marketing materials) and ship the application with language strings for that language only.

Test Methodology

During the in-lab review, the test vendor will examine your source code for hard-coded strings.

Criteria for Passing

This is a recommendation only. Failure to comply with this recommendation will not cause the application to automatically fail the test.

Technology Configurations and Platform Requirements

Your application must meet the following technology and platform requirement:

- [Requirement 5.1: An ISV application must support the infrastructure that Microsoft Dynamics SL supports](#)

7.1 An ISV Application Must Support the Infrastructure that Microsoft Dynamics SL Supports

Type	Test Method	Technology	Solution Category		
Required	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

Your application must run on the specified infrastructure (browser, database, operating system, and other software) versions that the version of Microsoft Dynamics SL being tested runs on, or on later versions. Additionally, you must not require any version of a technology that conflicts with those required for Microsoft Dynamics SL.

If the most recent service or feature pack for the local version of Microsoft Dynamics SL has been available for more than two months, your application must run on the latest service or feature pack version of Microsoft Dynamics SL.

Resources

For more information, refer to the *Microsoft Dynamics SL System Requirements Guide*, available on the PartnerSource Web site (<https://mbs.microsoft.com/partnersource/>).

How to Comply

Test your application on the infrastructure that Microsoft Dynamics SL supports. In your implementation or installation documentation, include a system requirements section that identifies the supported operating system(s), database(s), browser(s), and other client environment requirements. Specify the required version numbers for all required infrastructure software.

Test Methodology

The test vendor will perform a qualitative review to determine whether your application runs on the prescribed infrastructure (browser, database, operating system, and other required software). The test vendor will review your user documentation and compare the listed requirements to the latest list of supported components for Microsoft Dynamics SL.

Criteria for Passing

This requirement is mandatory. If the ISV solution does not run on the prescribed infrastructure, it will fail the test.

Installation and Setup Requirements

Your application must meet the following installation and removal (uninstall) requirements:

- [8.1 The ISV Application Must Include Documented Installation Procedures](#)
- [8.2 The ISV Application Installation Must Correctly Register DLLs and COM Components](#)
- [8.3 The ISV Must Document the Version and Service Pack of All Dependent Software Programs, Including Microsoft Dynamics SL, Required for Installing Its Application](#)
- [8.4 The ISV Must Document the Microsoft Dynamics SL Modules Required for Using Its Application](#)
- [8.5 The ISV Must Document the Procedures for Removing Its Application](#)
- [8.6 The ISV Application Should Include Installable Demonstration Data](#)
- [8.7 The ISV Application Must Not Adversely Affect Microsoft Dynamics SL After It Is Installed](#)

8.1 An ISV Application Must Include Documented Installation Procedures

Type	Test Method	Technology	Solution Category		
Required	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

All ISV applications must have complete application installation instructions. The instructions must be clear and easy to follow. The installation instructions must include any necessary procedures for installing and configuring Microsoft Dynamics SL that are not already part of the *Microsoft Dynamics SL Installation Guide* so that it functions with the ISV application. The instructions must be part of the standard user documentation and must list all necessary steps, including working with the Microsoft Dynamics SL product, system settings, and instructions for using any automated installation executables.

In addition to this requirement, you should provide a list of all files included with your installation. This list should include all executables (.EXE), reports (.RPT), stored procedures (.CRP), create table statements (.CRT), create index statements (.CRU), statements to include new values in existing tables, and possible values (.CSV) files. You should provide references to these files in your installation instructions. Users should be able to rerun SQL scripts without generating errors.

Resources

For more information, refer to the *Microsoft Dynamics SL Installation Guide*, located on the Microsoft Dynamics SL product DVD

How to Comply

You must provide instructions for installing the ISV application.

Test Methodology

The test vendor will confirm that the ISV has provided installation procedures and a complete list of all resources added to the Microsoft Dynamics SL application. The test vendor will compare your component list and distribution media to confirm that all components are included.

The test vendor will follow each step in the installation instructions in the order presented. The test vendor must be able to complete the installation without consulting support personal or contacting the ISV.

Criteria for Passing

This requirement is mandatory. If the application does not include clear installation instructions, it will fail the test.

8.2 The Installation Process for an ISV Application Must Correctly Register DLLs and COM Components

Type	Test Method	Technology	Solution Category		
Required	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

Your application should have its own installation wizard.

The Setup program for both Microsoft Dynamics SL–based and .NET Framework–based applications should record any DLLs and ActiveX components in the registry database of the operating system. The registry serves as a central configuration database for user, application, and computer-specific information.

If your application installs any dynamic-link libraries (DLLs) or component object model (COM) components including ActiveX controls, you must provide a Setup program for your application. Users should not have to manually copy files or run SQL scripts; these operations should be performed by the installation process. The Setup program must record the COM components in the registry database of the operating system. The registry serves as a central configuration database for user, application, and computer-specific information. In addition, the Setup program should place shared .NET components in the Global Assembly Cache.

Resources

For .NET Framework–based applications, Microsoft recommends the use of the Windows Installer, an application installation and configuration service. The Windows Installer is an operating system component that centrally manages application installation configuration and application removal that lets the operating system manage application setup and configuration.

For more information about Windows Installer, see [Overview of the Windows Installer Technology](#). For information about the Global Assembly Cache, see [Global Assembly Cache](#) and [Working with Assemblies and the Global Assembly Cache](#).

How to Comply

Check the registry to make sure that your Setup program functions correctly. Document the correct registry settings and include this information with your application when you submit it for testing.

Test Methodology

The test vendor will confirm that the ISV has provided a complete list of all resources added to the Microsoft Dynamics SL application. The list will also be used to verify the removal of the product. During

the in-lab test, the test vendor will install your application according to the instructions from requirement 1.2 and review the registry to verify that the Setup program registers all COM components.

Criteria for Passing

This requirement is mandatory. If the necessary COM components are not registered correctly, the application will fail the test.

8.3 The ISV Must Document the Version and Service Pack of All Dependent Software Programs, Including Microsoft Dynamics SL, Required for Installing Its Application

Type	Test Method	Technology	Solution Category		
Required	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

ISV applications can have external software dependencies. Additionally, Microsoft Dynamics SL requires specific software and service pack versions to be installed. You must document these requirements and include the documentation in your test submission.

Other than external product dependencies, you must provide all files required to install your application completely. Users should not be required to purchase other components from you or download additional items other than documentation from your Web site.

Note: You may distribute feature upgrades or additional modules to your product, with or without charge, via your Web site, but these upgrades and modules must not be core components that are required in order for your solution to function properly.

Resources

None

How to Comply

You must provide the test vendor with a list of the software, including version numbers and service pack levels, that your application requires. The list must not include additional components that must be purchased from you or downloaded from your Web site. An exceptions can be granted for documentation for connected solutions installed on remote devices as described in 2.1 An ISV Application Must Include User Documentation.

Test Methodology

Your application installation instructions must not require the purchase of other components from you, nor must they direct the user to a Web site for additional component downloads. All components required to launch and operate the ISV solution must be readily available on the application distribution media that you provide to customers. The test vendor will review the documentation to verify that the required software list is provided and that it does not contain any mandatory component purchases or downloads from your site.

Criteria for Passing

This requirement is mandatory. If the ISV does not document the software and service pack requirements, the application will fail the test.

8.4 The ISV Must Document the Microsoft Dynamics SL Modules Required for Using Its Application

Type	Test Method	Technology	Solution Category		
Required	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

If your application requires certain Microsoft Dynamics SL modules to be installed, you must document these requirements.

Resources

None

How to Comply

You must provide the test vendor with a list of the Microsoft Dynamics SL components that your application requires.

Test Methodology

The test vendor will review the documentation to verify that the required software list is provided.

Criteria for Passing

This requirement is mandatory. If the ISV does not provide a list of required Microsoft Dynamics SL components, the application will fail the test.

8.5 The ISV Must Document the Procedures for Removing Its Application

Type	Test Method	Technology	Solution Category		
Required	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

Customers must be able to uninstall an ISV application and continue using Microsoft Dynamics SL. Therefore, you must provide instructions for removing your application. These instructions must include procedures for removing any imported code, removing the application itself, and removing registry information for any .DLL or ActiveX components. It is preferred that the removal be fully automated. However, this it is not required as long as all manual steps are documented.

You must also include procedures for restoring Microsoft Dynamics SL to its state before the ISV application was installed. After the application is removed from the system, the Microsoft Dynamics SL application must still function as it did before your application was installed.

Resources

None

How to Comply

First, automate as much of the uninstall process as possible. You must provide a complete list of all resources that the ISV application adds to the Microsoft Dynamics SL application environment. You must provide instructions for installing and uninstalling the ISV application, including removing any code, DLL and ActiveX components, and registry entries.

Test Methodology

The test vendor will confirm that the ISV has provided a complete list of all resources (files and registry entries) added to the Microsoft Dynamics SL application environment. The test vendor will use this list to verify that your application is completely removed from the Microsoft Dynamics SL environment.

The test vendor will follow each step in the removal instructions in the order presented. The test vendor must be able to remove the product without consulting the ISV. After removal of the application, Microsoft Dynamics SL must be fully functional and all resources must be back to their original state. The test vendor will review the list of components and registry entries that you provided to verify that all portions of your application have been removed.

The test vendor will review the list of components that the ISV application installed to verify that the entire ISV solution has been removed.

Criteria for Passing

This requirement is mandatory. If the application does not uninstall completely or if Microsoft Dynamics SL no longer functions after the application is removed, it will fail the test.

8.6 An ISV Application Should Include Installable Demonstration Data

Type	Test Method	Technology	Solution Category		
Recommended	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

Demonstration data is useful for many purposes, such as sales demonstrations, training, and application testing. Therefore, each ISV should deliver at least one demonstration company with its application or integrate its data with the current Microsoft Dynamics SL demonstration data and supply instructions that describe how to add the demonstration data to Microsoft Dynamics SL.

Microsoft recognizes that some data centers do not permit installation of demonstration data on production servers. For this reason, you can deliver demonstration data as part of the main installation or as a separate installation (for example, virtual PC [VPC] image).

Resources

None

How to Comply

The demonstration data should populate the new tables or fields in your application. Include complete instructions for adding the data to Microsoft Dynamics SL, if necessary.

Test Methodology

To verify this requirement, after installing the ISV application, the test vendor will review the demonstration data.

Criteria for Passing

This is a recommendation only. Failure to comply with this recommendation will not cause the application to automatically fail the test.

8.7 An ISV Application Must Not Adversely Affect Microsoft Dynamics SL After It Is Installed

Type	Test Method	Technology	Solution Category		
Required	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

The specific intent of this requirement is to eliminate poor customer experiences that result from new installations that produce errors the first time an application is started. After the ISV application is installed, Microsoft Dynamics SL must start without any error messages or system crashes.

Resources

None

How to Comply

Your installation guidance or program must be sufficient to avoid these errors.

Test Methodology

The test vendor will install your application, and then start Microsoft Dynamics SL. The vendor will then log on to the system as system administrator (SYSADMIN) or as a user. The system must not display any error messages when Microsoft Dynamics SL starts or during the logon process.

After installation, the test vendor will inspect for a problem-free Microsoft Dynamics SL launch. The test vendor will test basic functionality to ensure no errors are encountered when Microsoft Dynamics SL starts or during the logon process.

Criteria for Passing

This requirement is mandatory. If an error is encountered in Microsoft Dynamics SL after the ISV installation is complete, it will fail the test.

Backup and Restore

Your application must meet the following backup and restore requirement:

- [9.1 The ISV Must Include Procedures to Back Up and Restore Its Application and Data](#)

9.1 The ISV Must Include Procedures to Back Up and Restore Its Application and Data

Type	Test Method	Technology	Solution Category		
Required	In-lab test	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

A customer or partner must be able to back up and restore your application and all associated data. Therefore, if your application requires any special back up or restore procedures, you must include documentation that describes what should be backed up, how to back it up, and how to restore data.

Backup procedures should also include any application data stored outside Microsoft Dynamics SL.

Applications that store some data outside of the Microsoft Dynamics SL might require this additional documentation.

Resources

None

How to Comply

If your application requires the user to perform any special steps to back up or restore data compared to a standard Microsoft Dynamics SL installation, you must prepare a document that describes the backup process. Include the following:

- What to back up
- How to back it up
- How to restore the data, including data stored outside Microsoft Dynamics SL

Test Methodology

During the in-lab test, the test vendor will verify that you have included appropriate backup and restore procedures. The test vendor will perform the backup and restore process to make sure that it functions correctly (for example using your demonstration data).

Criteria for Passing

This requirement is mandatory. If the ISV solution does not include appropriate backup and restore procedures, it will fail the test.

Extensibility and Customization Requirements

Your application should meet the following extensibility and customization recommendation:

- [10.1 The ISV Should Document Its Application Extensibility and Customization Strategy](#)

10.1 The ISV Should Document Its Application Extensibility and Customization Strategy

Type	Test Method	Technology	Solution Category		
Recommendation	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

Regardless of the capabilities of your application, customers and value-added resellers (VARs) frequently customize and extend business software. Embedded solutions should be extensible and customizable by partners and customers seeking to meet their unique requirements using the Microsoft Dynamics SL Customization Manager module. Customization capabilities include the ability to add or hide fields, make fields required or disabled, and supply default values. A customer or partner must have the information required to perform these customizations successfully so that the application continues to function as designed. Therefore, you should provide documentation that explains how to customize your application.

Resources

- Microsoft Dynamics SL Software Development Kit (SDK) (SL_SDK.PDF)
- Microsoft Dynamics SL Customization Manager Guide (SL_CU.PDF)

How to Comply

Document your customization and extensibility procedures in a developer's guide. You should provide an overview that explains the customization and extensibility strategy, as well as detailed information about each API, Web service, and so on, that your application exposes. If your application uses the capabilities of the Microsoft Dynamics SL Customization Manager module, you can refer to the appropriate documentation provided with Microsoft Dynamics SL.

Test Methodology

During the in-lab test, the test vendor will verify that you have included a developer's guide that includes an overview and detailed information.

Note: The test vendor will validate that the required information is included. The test vendor will not test the procedures or verify the quality of the information.

Criteria for Passing

This is a recommendation only. Failure to comply with this recommendation will not automatically cause the application to fail the test.

Upgrade and Maintenance

Your application must meet the following upgrade and maintenance requirements:

- [11.1 The ISV Must Provide Database Upgrade Scripts](#)

- [11.2 The ISV Must Use File Versioning for DLLs and COM Components](#)

11.1 The ISV Must Provide Database Upgrade Scripts

Type	Test Method	Technology	Solution Category		
Required*	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

*If the application has been updated.

Summary and Intent

Upgrades are an important part of business software. The intent of this requirement is to make upgrades easier for partners by providing scripts that support database upgrades.

Resources

None

How to Comply

Prepare for this requirement by preparing and documenting your upgrade scripts. Your documentation must, at a minimum, list the names of the upgrade scripts and the tables that each script affects. More in-depth documentation will be helpful to everyone who uses your application, but is not required.

Test Methodology

During the in-lab test, the test vendor will verify that you have included the required upgrade script documentation, which must list the scripts and the affected tables.

Criteria for Passing

This requirement is mandatory. If the ISV does not provide and document the upgrade scripts for an upgraded application, the application will fail the test.

11.2 The ISV Must Use File Versioning for DLLs and COM Components

Type	Test Method	Technology	Solution Category		
Required	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

All executable files, such as DLLs and COM components (including ActiveX controls), must have versioning metadata associated with them. Installation programs use this metadata to confirm that correct versions are in place before applying an upgrade, service pack, or hot fix. Without this versioning information, an installation program could corrupt the system by applying changes that cannot synchronize with the file that is currently installed. Additionally, if a shared component fails, correct file version information lets a customer identify the associated application and file producer. The file's

producer is the only entity that can regress the file; therefore, the metadata must also include the company name.

Resources

None

How to Comply

Before you submit your application for testing, examine the file information for each DLL and COM component to verify that it includes the product name, company name, and file version number.

Test Methodology

The test vendor will verify that installed components contain metadata with the following elements:

- Product name
- Company Name
- File Version Number

Criteria for Passing

This requirement is mandatory. If the ISV does not use file versioning, the application will fail the test.

Best Practice Guidelines

The following best practices are strongly recommended for use by ISVs but they are not part of the test process.

Trustworthy Computing Best Practices

Your application should comply with the following trustworthy computing requirements:

BP1 The ISV Must Educate Staff on the Security Development Lifecycle Before Starting Design and Development Work

Type	Test Method	Technology	Solution Category		
Required	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

Microsoft has adopted a process called the *Trustworthy Computing Security Development Lifecycle (SDL)* to help ensure that software development follows security best practices. Security best practices require that developers are aware of secure coding practices including threats and countermeasures, and that they use the most recent versions of the Microsoft Visual Studio® development system, .NET Framework, ADO.NET, and other tools and technologies. These tools provide the greatest compatibility with the platform infrastructure and provide the latest security advancements.

In addition, SDL requires that developers fix all critical bugs that compromise security and perform security code reviews based on guidelines and checklists described in Appendix D of *Writing Secure Code* by Michael Howard and David C. LeBlanc. These reviews help ensure that the software design meets minimal trustworthy computing standards.

Resources

- [Microsoft Trustworthy Computing Web site](#)
- [Writing Secure Code, Second Edition by Michael Howard and David LeBlanc](#)
- [Companion content for Writing Secure Code, Second Edition](#)

How to Comply

To satisfy the minimum education requirement, all developers must do the following:

- Read *Writing Secure Code, Second Edition*, by Michael Howard
- Complete the following two Microsoft eLearning security courses:
 - [Clinic 2806: Microsoft Security Guidance Training for Developers](#)
 - [Clinic 2807: Microsoft Security Guidance Training for Developers II](#)

To verify that your staff has satisfied this requirement, prepare checklists or training documentation (such as training overviews, Microsoft PowerPoint® presentations, class handouts, or syllabi) and include these items in your test submission package.

Test Methodology

The test vendor will review your training documentation to verify that your staff has completed the appropriate training. The vendor will use a scorecard similar to the following table.

	Developer/Tester Name	Date Completed
Writing Secure Code		
Clinic 2806		
Clinic 2807		
Other (specify)		
Other (specify)		

Criteria for Passing

This requirement is mandatory. If the ISV personnel do not complete the mandatory security training and follow the security guidelines, it will fail the test.

BP2 The ISV Must Establish and Follow Secure Development Best Practices

Type	Test Method	Technology	Solution Category		
Required	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

Your application development team must establish, communicate, and follow specific best practices for developing secure code. This approach will help them detect and remove security defects as early as possible in the development process. A number of resources, tools, and processes are available for you to use to accomplish this goal. The time and effort you take to apply best practices early in the product development cycle can prevent a more costly response to security defects later in the development cycle or even more painful responses after product release.

Resources

For more information, see the following:

- [Threat Modeling by Frank Swiderski and Window Snyder](#)
- [MSDN Library: Creating Named Shared Memory](#)

How to Comply

For .NET Framework–based applications, follow these guidelines:

- C# compiler: Use the latest version of the Microsoft Visual Studio® .NET development system. Use the specified compiler and linker flags for .NET Framework–based applications.
- csc.exe: Use version 8.0.50727.42.
- .NET Framework: Use version 2.0.50727 or higher.
- FxCop: Use the latest version of FxCop to locate and fix all security-related issues.

- Follow the guidelines in the SDL Shared section (that is, have no writable PE shared sections).
- Establish the following best practices as you develop your application:
 - Build tools: Use the currently required (or later) versions of compilers and compile options for your development platform.
 - Code analysis tools: Use the currently required (or later) versions of code analysis tools for native (C and C++) or managed (C#) code.
 - Previously released applications that were compiled with the **/GS** switch using Visual Studio version 2005 or later: You should research and resolve and **/GS** switch crashes. System crashes produced by binaries compiled with the **GS** option are marked in a specific way when uploaded through the Watson service. The data uploaded to Microsoft indicates where the buffer overflow occurred in the Microsoft code and confirms that it happened in a customer use scenario. Because the **GS** compiler option mitigates some types of buffer-overflow exploits, but not all of them, it is very important that you fix any reported buffer overruns in your code. For more information, see [Writing Secure Code, Second Edition \(Howard\)](#) and [SDL Buffer Overflow Watson Crash Requirements](#), which outlines the process and requirements for servicing **GS** crashes.
 - Banned APIs: New native code (C and C++) should not use banned versions of string buffer handling functions. Based on analysis of previous MSRC cases, you should avoid the use of certain APIs to reduce vulnerability. See [Writing Secure Code, Second Edition \(Howard\)](#) for more information.
 - Executable pages: Avoid using executable pages. Unless your technology requires executable pages (for example, just-in-time [JIT] compilers), do not use them. This requirement affects how you call the **VirtualAlloc()** APIs and prohibits compiling with **/NXCOMPAT:NO**. This requirement applies to Win32 and Win64 applications only; it does not apply to Windows Mobile.)
 - Writable shared PE sections: Your shipped binaries should not contain sections marked as **shared**. Shared binaries are a potential security threat. Use properly secured dynamically created shared memory objects instead.

Test Methodology

The test vendor will verify that your application meets the best practices requirements as follows:

- Tools and processes: The vendor will perform a qualitative review to make sure that required processes and tools are in place and that processes were successfully executed. (The vendor will not audit the output of your tools and processes.)
- FxCop: The vendor will run FxCop and will require that you fix all reported issues.
- .NET Framework SDL Shared Section requirements (no writable PE shared sections): The vendor will run tools and require that you fix all reported issues.

Criteria for Passing

This requirement is mandatory. If the ISV solution does not follow security guidelines, it will fail the test.

Design and Development Best Practices

BP3 The ISV Should Establish a Repeatable Process for Creating and Reviewing Design Specifications

Type	Test Method	Technology	Solution Category		
Recommendation	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

Software design is a complex process that involves market research into customers' needs and technical research into the feasibility of leveraging the latest advances in technology. You should establish a process that accommodates the iterative nature of designing applications.

By constantly evaluating and refining your design process, you will produce a "recipe" for designing stable and usable applications. The recipe will become a repeatable process, which will enable you to produce applications more efficiently and effectively, and with less rework.

Resources

- MSDN article ["Design Skills: The Practice of Design"](#)

How to Comply

- Create a template for your design specification that includes a description of the target market for the application, a list of reviewers, and the specification's history.
- Include sections that address the needs of the following disciplines and invite representatives from each of the disciplines to participate in the review process:
 - Development
 - Test
 - User Experience (UI design)
 - User Assistance (UI text and documentation)
 - Localization (if needed)
 - Build
 - Release
 - Sustaining engineering

In addition, input from the following teams will be helpful and should be solicited by the prototype phase at the latest: Customers, Partners, Support, Product Planning, Marketing, and General Management (executive sponsor).

- Make all revisions to the specification with "track changes" turned on to keep track of all updates to the specification.
- Periodically review the specification template and update it based on key learnings from previous releases.

Test Methodology

The test vendor will review the application's design specification to verify that the appropriate disciplines were represented in the design creation and review process and that revisions were noted in the history section of the specification. The test vendor will not compare the contents of the design specification with the application's functionality to verify that the finished product complies with the documented design.

Criteria for Passing

This is a recommendation only. Failure to comply with this recommendation will not automatically cause the application to fail the test.

BP4 The ISV Should Implement a Bug Tracking System

Type	Test Method	Technology	Solution Category		
Recommendation	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

Discovery of software bugs is an expected part of software development. Building security and quality into a software product entails tracking the bugs so that they are recorded in a database that can be accessed by developers and testers. Bug tracking systems help ensure that bugs, including "design bugs," are not overlooked, even if they must be deferred for a hot fix, service pack, or future release.

Resources

- Visual Studio 2005 Technical Article ["Version control, work item tracking, project management, and build management"](#)

How to Comply

Many development tools packages include a bug tracking system that is designed to work with the development tools. If your development environment does not include bug-tracking software, you should purchase a commercially available product or build your own. The types of information that you should be able to capture about a bug include the following:

- Unique bug ID
- Date that the bug was entered into the system
- Description of the bug and the steps to reproduce it
- File attachments (screen shots, script files, etc.)
- Severity – how serious is the impact of the bug? A bug that results in data loss or corruption has a higher impact on product quality and stability than a cosmetic bug.
- Priority – how urgently does the bug need to be fixed? A low-impact bug that blocks further development or testing progress must be fixed immediately, even before more severe non-blocking bugs.
- Assigned to – developer, tester, build queue, etc.
- Documentation impact – affects user documentation, affects release notes, etc.

- Status – open, resolved, closed, deferred, etc.
- Resolution – fixed, duplicate of another bug, unable to reproduce, acceptable workaround exists, etc.
- Date or build number that the bug is fixed in or closed

Test Methodology

The test vendor will review a sample bug report to verify that you are capturing the appropriate information in your bug tracking system. The test vendor will not review individual bugs.

Criteria for Passing

This is a recommendation only. Failure to comply with this recommendation will not automatically cause the application to fail the test.

BP5 The ISV Should Remove Non-Functioning Code from the Application’s Code Base

Type	Test Method	Technology	Solution Category		
Recommended	In-lab review	All	Simple	Complex	Hosted
			✓	✓	✓

Summary and Intent

Non-functioning code is confusing to support engineers who are attempting to resolve a problem and to developers who may work on your application in the future. In addition, there is always the risk, however remote, that code artifacts can be started accidentally by another application or process.

Resources

[How to: Obtain Code Coverage Data](#) describes how to configure test runs to produce code coverage data. You can use this data to remove unused code.

How to Comply

It is a good practice to remove all non-functioning code from an application before the application is released. Your test plans and test scripts should check for non-functioning code.

Test Methodology

During the in-lab review, the test vendor will run the best practice tool to determine if non-functioning code has been removed.

Criteria for Passing

This requirement is mandatory. If the application contains non-functioning code, it will fail the test.

Appendix A: User Interface (UI) Guidelines

The following best practices are provided to help ISVs develop applications that have a “look and feel” consistent with Microsoft Dynamics SL. As such, many best practices are more applicable to embedded solutions than to external applications.

Application Windows

The screens in an embedded Microsoft Dynamics SL application should comply with the following guidelines:

- UI 1.1 – Do not change the window size when your application opens a new window. For example, the window size must not change if a user changes from form view to grid view. (A user should be able to use the standard Microsoft Windows operating system tools and conventions to resize windows.)
- UI 1.2 – Make sure that application windows fit into an SVGA display with 1024 x 768 pixels.
- UI 1.3 – By default, do not require horizontal scrolling for application windows.
- UI 1.4 – Do not use absolute coordinates to specify the size and location of a new window. When a window opens, it should automatically size itself without exceeding preset size limits for windows.
- UI 1.5 – Do not use third-party grid controls in screens.
- UI 1.6 – If a screen or dialog box starts a process, make sure that it contains an **OK** button and a **Cancel** button.
- UI 1.7 – In a screen, make sure that the **OK** and **Cancel** buttons are the right-most buttons on the last line in the window.
- UI 1.8 – If you have additional command buttons to the left of the **OK** and **Cancel** buttons, separate them from the **OK** and **Cancel** buttons.
- UI 1.9 – Make sure that **OK** and **Cancel** buttons have exactly these labels. Do not use variant label forms such as **Ok** or **Continue** and **Quit**.

Edit Controls

The edit controls in a Microsoft Dynamics SL application should comply with the following guidelines.

Note: The following guidelines for edit controls also apply to read-only fields because a read-only field is an edit control where editing has been blocked.

- UI 2.1 – Do not override the default edit control labels if the new value is identical or similar to the Microsoft Dynamics SL standard label.
- UI 2.2 – Do not use fixed labels. To help with localization, obtain all static text in a window (headings, instructions, edit control labels, and so on) from the Microsoft Dynamics SL database using the Customization Manager.
- UI 2.3 – Do not use skip fields. The user must be able to modify all edit controls or the fields must be read-only.
- UI 2.4 – Do not use dimmed text in edit control labels.

- UI 2.5 – Use ordinary text in text boxes. Do not use dimmed text, bold, italic, and so on.
- UI 2.6 – Use positive statements for check box labels. Do not use negative statements because some users might not understand the implied double negative that occurs when a negatively labeled check box is cleared. For example, do not use the label **Don't tell me again** for a check box. Instead, use the label **Display this message in the future** and change the default value to **false** (unchecked).
- UI 2.7 – Do not use check boxes that contain mutually exclusive options. Instead, use drop-down lists or option (radio) buttons.
- UI 2.8 – Use drop-down lists instead of option buttons for mutually exclusive choices.

Buttons

The buttons in a Microsoft Dynamics SL application should comply with the following guidelines.

- UI 3.1 – Make sure that a user receives visible feedback when he or she clicks a button.
- UI 3.2 – If a button opens a new window, use the button label as the prefix for the title of the new window.
- UI 3.3 – In forms, put command buttons on the right side or at the bottom of the window.
- UI 3.4 – If you use buttons for Details and Inquiries, use exactly these names for the buttons.
- UI 3.5 – If you use Details and Inquiries buttons, put the buttons in exactly this order.
- UI 3.6 – If you include a Details button, make sure that it is the first button.
- UI 3.7 – Do not use command buttons to invoke a dialog box. Instead, make sure that the module performs the only possible action as soon as the user clicks the button unless the action removes data from the database or affects the operation of other windows.
- UI 3.8 – Use text for command button labels. Do not use pictures or symbols.
- UI 3.9 – Make sure that users must click a button only once to perform the associated action.
- UI 3.10 – Use icon buttons sparingly. Add a tooltip to describe the button's use and place the button immediately to the right of a text box.

Other Controls and Toolbars

Other controls and toolbars in a Microsoft Dynamics SL application should comply with the following guidelines.

- UI 4.1 – Use consistent names and organization.
- UI 4.2 – Use consistent names in all modules. For example, do not name a button **Inquiries** in one module and **Inquiry** in another module.
- UI 4.3 – Use TAB sequences that are natural from the user's point of view. The TAB sequence is the order in which the pointer travels from control to control when a user presses the TAB key. The natural order is usually column-to-column.
- UI 4.4 – Use dimmed text only to indicate unavailable buttons or menu items.
- UI 4.5 – Do not use hidden controls. Deactivate them if they are not relevant in the current context.

- UI 4.6 – You should not use toolbars that duplicate functions that are available in the standard Microsoft Dynamics SL toolbar.

Icons and Symbols

Icons and symbols in a Microsoft Dynamics SL application should comply with the following guidelines.

- UI 6.1 – Make sure that a typical user can understand all icons and symbols.
- UI 6.2 – Make sure that the icon style complies with Microsoft Office guidelines.
- UI 6.3 – For each icon, include a screen tip that explains the purpose of the icon.
- UI 6.4 – Use color in icons and symbols. Do not use icons that contain only black, gray, and white.
- UI 6.5 – Use a transparent background for the icons.
- UI 6.6 – Do not use text or characters in icons and symbols because such icons and symbols are difficult to localize.
- UI 6.7 – Do not use blinking icons or symbols.

User Assistance

Microsoft Dynamics SL supports the following guidelines for user assistance:

- **Tasks** – Task help appears when users press **F1** and then browse the **Contents** pane for “How do I...?” topics. Task help provides information about how to complete a user procedure. Task help takes the user’s point of view and should use second person singular (for example, “You do this”) or imperative (for example, “Do this...”) voice.
- **Concepts** – Conceptual help topics describe typical business processes and how the application performs the processes. Conceptual help is informational and can use the third person (for example, “The system does this...” or “Users can do that”).
- **Reference** – Reference help provides detailed information about an individual screen. (In contrast, screen tips, tool tips, and status bar messages provide detailed information about a single screen object.) Reference help should describe each UI element without repeating the label or field caption. It should also inform the user about where the value will be used in downstream processing.
- **Tool tips** – A tool tip appears next to a command button (including a toolbar button) after a brief delay when a user uses the mouse to hover over the button.
- **Status bar messages** – A status bar message automatically appears in the left side of the status bar (in the lower-left corner of the Microsoft Dynamics SL screen).

Help

Help in a Microsoft Dynamics SL application should comply with the following guidelines.

- UI 7.1 – In Reference help, explain the overall purpose of the screen, when and why the screen should be used, and how its use will affect the rest of the system.
- UI 7.2 – Provide Reference help for each screen.

Screen Tips and Tool Tips

Screen tips and tool tips in a Microsoft Dynamics SL application should comply with the following guidelines.

- UI 8.10 – In each screen tip, answer the questions “What is this?” and “Why should I use it?”
- UI 8.11 – Provide information that the object label does not convey. Duplicating an object label, even, for example, using a different word order, is not helpful.
- UI 8.12 – You do not have to provide screen tips for check boxes or radio buttons.

Messages

Messages in a Microsoft Dynamics SL application should comply with the following guidelines.

- UI 9.1 – Classify messages to the user as *critical*, *warning*, and *information* to comply with Windows User Experience rules.
 - A *critical* message informs the user of a serious problem that requires intervention or correction before work can continue.
 - A *warning* alerts the user to a condition or situation that requires a user decision and input before processing can continue, such as an impending action with potentially destructive, irreversible consequences. The message can be in the form of a question: for example, “Save changes to MyReport?”
 - An *information* message provides information about the results of a command. It offers no user choices. The user acknowledges the message by clicking the **OK** button.

Note: Do not confuse critical messages and warnings.

- UI 9.2 – Make sure that critical and warning messages are constructive. They must describe a way of solving the user’s problem.
- UI 9.3 – Make sure that a typical user can understand the message. Write the message clearly, without jargon or confusing terminology.
- UI 9.4 – In informative messages, include an **OK** button and no other buttons.
- UI 9.5 – In a message box, include **Yes** and **No** buttons only if the message contains a clearly phrased question to the user.
- UI 9.6 – Do not use routine confirmation messages. You may provide confirmation messages for transactions that users consider critical; for example, you could use them for password changes or account deletions.

Appendix B: Microsoft Dynamics SL Consistency Verification Test

Company:

Application:

Version:

Prepared by:

Date Prepared:

Identify the intended purpose of the product.

Specify what fundamental service the product is supposed to provide. To the extent possible, define the audience for the product. Write a paragraph that briefly explains the purpose of the product and describes its intended audience. Include whether the solution is embedded, external, or both.

Example:

Test Application is an external application for managing projects that enables the user to track the project status using a handheld device.

Describe the product purpose:

Identify primary functions.

Term	Definition	Notes
Primary function	Any function so important that, in the estimation of a typical user, its inoperability or impairment would make the product unfit for its stated purpose.	A function is <i>primary</i> if you can associate it with the purpose of the product and it is essential to that purpose. Primary functions define the product. For example, the function of adding text to a document in Microsoft Word is so important that the product would be useless without it. Groups of functions, taken together, could be a primary function. For example, although no single function on the drawing toolbar of Word is primary, the complete toolbar might be primary. If the toolbar is primary, most of the functions on that toolbar should be operable for the product to pass the test.

Examples of primary functions:

- Manage labor collection from employees who work on multiple projects across multiple sites.
- Manage stock environment.
- Manage IT interoperability with a fast carrier company.

List all primary functions:

Identify contributing functions.

Term	Definition	Notes
Contributing function	Any function that contributes to the utility of the product but is not a primary function.	Although contributing functions are not primary, their inoperability could be grounds for test failure. For example, users may be able to do useful things with a product, even if it has an Undo function that never works, but most users will find that unacceptable. Such a failure would violate fundamental expectations about how products should work.

Example of a contributing function:

- Generate a 3-D report.

List all contributing functions:

Specify potential instabilities and challenging data.

List 5-10 functions or groups of functions (preferably primary functions) for focused instability testing.

Specify challenging data for each selected function. Think of large, complex, or otherwise challenging input.

Examples:

- Functions that interoperate with other products (for example, object linking and embedding, file conversion).
- Functions that handle events external to the application (for example, wake up a sleeping computer when a fax arrives).
- Functions that make intensive use of memory.
- Functions that interact extensively with the operating system.
- Functions of unusual complexity.
- Functions that change operating parameters (for example, preference settings).
- Functions that manipulate operating system configuration.

List all potential instabilities and challenging data:

Design and record a consistency verification test.

Prerequisites:

List any action that must be performed before the consistency verification test can be executed.

Examples:

- Install Microsoft SQL Server 2005.
- Use radio frequency identification (RFID) to identify hardware.

List all prerequisites:

Required Information:

List any information that a user must know to perform the consistency verification test.

Examples:

- User must log on as User x.
- User must know the product serial number.
- User must know the account passwords.

List all required information:

Test Procedure:

Complete the following procedure to test each primary function of the application. You must describe each step that is required to test a primary function. You can combine similar functions, if appropriate.

Example:

- Manage cross-docking operations.
 1. Define the goods involved.
 2. Determine the delivery locations.
 3. Determine the transport company.
- Manage the stock environment.
 1. On the menu, select **File** and then select **Save**.
 2. Type the file name in the field, select the location for the file, and then click **OK**.
- Manage IT interoperability with a fast carrier company.
 1. Configure database access.
 2. Define user rights and circuit approval.

List all test procedures: