

Configuring the VCS Integration Feature

Innovation Release

Version 10.0

April 2017

This document applies to webMethods Integration Server and Software AG Designer Version 10.0 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2007-2017 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Table of Contents

About this Guide	7
Document Conventions.....	8
Online Information.....	9
Overview	11
About the VCS Integration Feature.....	12
Locking Files Locally versus VCS Locking.....	13
System Locking and the VCS Integration Feature.....	13
How Does the VCS Integration Feature Differ from Local Service Development?.....	13
Terminology.....	14
Architecture.....	15
Basic Concepts.....	16
Summary of VCS Commands.....	17
Hierarchical Operation.....	18
Typical Work Flow.....	19
Understanding Package, Element, and File Behavior.....	19
Location of a Package.....	19
Package Contents and Interaction with the VCS Integration Feature.....	20
..\packages\packageName Directory.....	20
\code Directory.....	20
\config Directory.....	21
\doc Directory.....	21
\lib Directory.....	21
\ins Directory.....	21
\pub Directory.....	24
\resources Directory.....	24
\templates Directory.....	24
\web Directory.....	24
Applying Labels within the VCS Repository.....	24
Working with Blaze Rule Services.....	25
Working with Web Service Descriptors.....	25
System Requirements	27
Platform, Hardware, and VCS Requirements.....	28
Visual SourceSafe Configuration.....	29
ClearCase Configuration.....	30
Subversion Configuration.....	30
Updating the Path Environment Variable For Subversion.....	31
Excluding File Types.....	31
Configuring the VCS Integration Feature	33
Overview.....	34

Configuring Integration Server and Designer.....	34
Removing the watt.server.ns.backupNodes Setting.....	34
Configuring Designer.....	34
Configuring the WmVCS Package to Work with Visual SourceSafe.....	35
Configuring the WmVCS Package to Work with ClearCase.....	36
Configuring the WmVCS Package to Work with Subversion.....	37
Connecting to Subversion for the First Time.....	38
Disconnecting from Subversion.....	38
Reconnecting to Subversion.....	38
Configuring Integration Server to Connect to a Subversion Repository through a Non-English Subversion Client.....	39
Configuring Access to the VCS Repository.....	39
Working with a Windows System.....	40
Creating VCS Users.....	40
Mapping One Integration Server User to One VCS User.....	40
Mapping Multiple Integration Server Users to One VCS User.....	41
Editing a User Mapping.....	41
Deleting a User Mapping.....	42
Enabling and Disabling the VCS Integration Packages.....	42
Temporarily Disabling the VCS Integration Feature.....	43
Enabling the VCS Integration Feature.....	44
Permanently Disabling the VCS Integration Feature.....	44
Removing the VCS Support Package.....	45
Working in a Shared Development Environment.....	47
General Behavior.....	48
Implementing a Shared Development Environment.....	49
Working with a Single Integration Server.....	49
Working with Multiple Integration Servers and VCS Servers.....	50
Working with Multiple Integration Servers and One VCS Server.....	50
Sharing Packages Among Integration Servers.....	51
Locking Files in a Shared Development Environment.....	51
Importing a Package from Another Integration Server.....	51
About the Manifest File.....	52
Behavior of the manifest.v3 File When Creating a New Package.....	52
Understanding Multiple User Accounts and VCS Integration.....	53
Security Considerations.....	55
General Security Considerations.....	56
About VCS Users.....	56
User Mappings for Visual SourceSafe and Subversion.....	56
VCS Commands.....	57
Logging.....	57
Checking of External Input.....	57
Validity of VCS History Information.....	58

Troubleshooting	59
Cannot Connect to the VCS Server.....	60
VCS Menu Commands Are Not Present.....	61
VCS Menu Commands Are Not Available.....	61
An "Access Denied" Exception Occurs When Saving an Element.....	62
Cannot Create a C/C++ Service.....	62
Cannot Get Latest Version of a C Service.....	62
Designer Loses Connection with Integration Server During Large Transactions.....	63
Designer Appears to Stop Responding After Applying a VCS Command.....	63
A Check-In Comment Is Not Recorded in the Revision History (Visual SourceSafe).....	64
VCS Transactions Result in Unexpected Behavior (Visual SourceSafe).....	65

About this Guide

This guide contains information for using the VCS Integration feature provided with webMethods Integration Server. This includes information for administrators who want to integrate with a version control system (VCS) and information for developers who want to create business integration elements using files that are maintained in an external version control system.

Important: The WmVCS package, which provides the functionality for using the VCS Integration Feature, is deprecated as of Integration Server version 9.9. Software AG recommends that you use the local service development feature (Local Version Control Integration) to check package elements and their supporting files into and out of a version control system (VCS) directly from Designer.

This guide contains the following:

- An introduction to the VCS Integration feature provided with webMethods Integration Server.
- Requirements for Integration Server to work with your VCS.
- Steps to configure the VCS Integration feature with Software AG Designer, Visual SourceSafe, ClearCase, and Subversion. As well as, steps to configure access to a VCS repository.
- Information about enabling and disabling the VCS package.
- Detailed information about implementing and working in a shared development environment.
- Security considerations.
- Troubleshooting tips.

This guide serves as a companion guide to the VCS Integration feature documentation provided in the Software AG Designer *webMethods Service Development Help*, which contains conceptual and procedural information about:

- Using the local service development feature with a VCS.
- Checking elements in and out of a VCS.
- VCS Integration feature support.
- Locking files locally, system locking, and locking elements.
- Adding, modifying, and deleting packages and elements in the VCS.
- Working with rules, adapter connections, and Java services.

Note: This guide describes features and functionality that may or may not be available with your licensed version of webMethods Integration Server. For information about the licensed components for your installation, see the **Settings > License** page in the webMethods Integration Server Administrator.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies storage locations for services on webMethods Integration Server, using the convention <i>folder.subfolder:service</i> .
UPPERCASE	Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+).
<i>Italic</i>	Identifies variables for which you must supply values specific to your own situation or environment. Identifies new terms the first time they occur in the text.
Monospace font	Identifies text you must type or messages displayed by the system.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at "<http://documentation.softwareag.com>". The site requires Empower credentials. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

You can find product information on the Software AG Empower Product Support website at "<https://empower.softwareag.com>".

To submit feature/enhancement requests, get information about product availability, and download products, go to "[Products](#)".

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the "[Knowledge Center](#)".

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at "<http://techcommunity.softwareag.com>". You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

1 Overview

■ About the VCS Integration Feature	12
■ Terminology	14
■ Architecture	15
■ Basic Concepts	16
■ Understanding Package, Element, and File Behavior	19

About the VCS Integration Feature

Software AG Designer enables you to create, maintain, and manage custom integration packages for use by webMethods Integration Server. Often, many enterprise organizations employ a version control system (VCS) for the development of software solutions, providing automatic auditing, versioning, and security to software development projects. Such products include Microsoft Visual SourceSafe, IBM Rational ClearCase, and Subversion, the open-source VCS.

In addition, the use of a shared VCS repository greatly enhances team development of software solutions. The Version Control System Integration feature (VCS Integration) enables you to interact directly with a resident VCS repository in the following ways:

- Check in
- Check out
- Revert changes
- Delete
- Get latest version
- Get earlier version
- View history

The VCS Integration feature also provides these features:

- Hierarchical behavior - VCS commands applied to packages and folders are applied to all of the folders and elements they contain.
- Pre-packaged VCS support for products such as Visual SourceSafe, ClearCase, and Subversion.
- Multiple Integration Server support - Multiple Integration Server installations can interact with a single, shared VCS server, enabling shared development across two or more servers.
- Multiple user support - Two or more Designer users can interact with the VCS server while working on the same Integration Server.
- Password-protected access to the VCS.
- Standard copy and paste, move, and rename behavior -when you use Designer to apply these actions to any package, folder, or element subject to VCS control, the VCS Integration feature automatically updates the VCS repository to reflect the changes.

The VCS Integration feature is intended for the control of webMethods packages and their contents in development environments only. The feature does not support:

- The ability to "diff" or merge files.

- The ability to move checked out elements loaded from an earlier VCS version.
- Static viewing of previous versions.
- VCS integration of files outside the package namespace structure (that is, components that do not appear in the Package Navigator view in Designer, or exist outside the `..\packages\ns` or the `..\packages\code\source` directories).
- VCS integration of backup files generated by Designer or Integration Server.

Locking Files Locally versus VCS Locking

In a shared development environment, there is typically a mechanism for a developer to lock a file during modification, and to unlock it when the modifications are complete. This prevents other developers from working on the file at the same time.

Designer and Integration Server provide basic locking and unlocking of project files on the Integration Server only, with no built-in interaction with an external VCS. In this case, Integration Server files have to be checked in to and out of the VCS manually, outside of the Integration Server or Designer.

For more information on the details of basic locking operation, see *webMethods Integration Server Administrator's Guide* and the *webMethods Service Development Help* in Designer.

The VCS Integration feature extends and is fully compatible with the basic locking functionality of Integration Server and Designer. When the VCS Integration feature is not installed or is disabled on an Integration Server, only the basic locking and unlocking functionality will be available in Designer.

System Locking and the VCS Integration Feature

Designer and Integration Server support the concept of *system locking*. When an element's server-side files are marked read-only on the Integration Server, the element is *system locked*. Files are generally system locked when a server administrator accesses the file through the server's operating system and marks the files as read-only. You cannot edit an element until the server administrator makes the element's server-side files writable and you reload the package in which the element resides.

When an Integration Server has the VCS Integration feature enabled, system locking is effectively disabled for elements that are checked into the version control system with the VCS Integration feature. The VCS Integration feature will override any read/write status changes applied manually by a server administrator.

How Does the VCS Integration Feature Differ from Local Service Development?

The VCS Integration feature is no the same as the local service development feature available in Designer. The two features differ as follows:

- The local service development feature is installed as a plug-in to Designer, whereas the VCS Integration feature is provided by way of the WmVCS Integration Server package. When a new version of your VCS client is released, with the local service development feature, you would need to reinstall the VCS client plug-in but you do not need to upgrade Designer. With the VCS Integration feature, you would need to update the WmVCS package every time a new version of a supported VCS client is released.
- Configuration for the local service development feature is performed in Designer rather than on Integration Server.
- The local service development feature permits Designer to connect to multiple VCS clients. The VCS Integration feature permits Integration Server to connect to only one of the supported VCS clients.
- The local service development feature performs version control tasks locally within the Eclipse framework. That is, commands are sent directly between Designer and the VCS client. Conversely, the VCS Integration feature performs version control tasks through Integration Server.
- The local service development feature uses the VCS client's menus and commands, which you may already know. Conversely, the VCS Integration feature uses its own commands to access the VCS client and therefore requires a little extra time to learn.
- Each feature supports a different set of VCS clients. The local service development feature supports Concurrent Versions System (CVS), Subversion (SVN), and Microsoft Team Foundation Server. The VCS Integration feature supports Microsoft Visual SourceSafe, IBM Rational ClearCase, and SVN and includes an exposed Java API that you can use to develop a connection to other VCS clients.
- With the local service development feature, you can check in all contents of a package, including Integration Server assets and other items that Integration Server does not maintain, such as HTML files. The VCS Integration feature requires you to manually check in items that Integration Server does not maintain.

For more information about using local service development in Designer, see the *webMethods Service Development Help*.

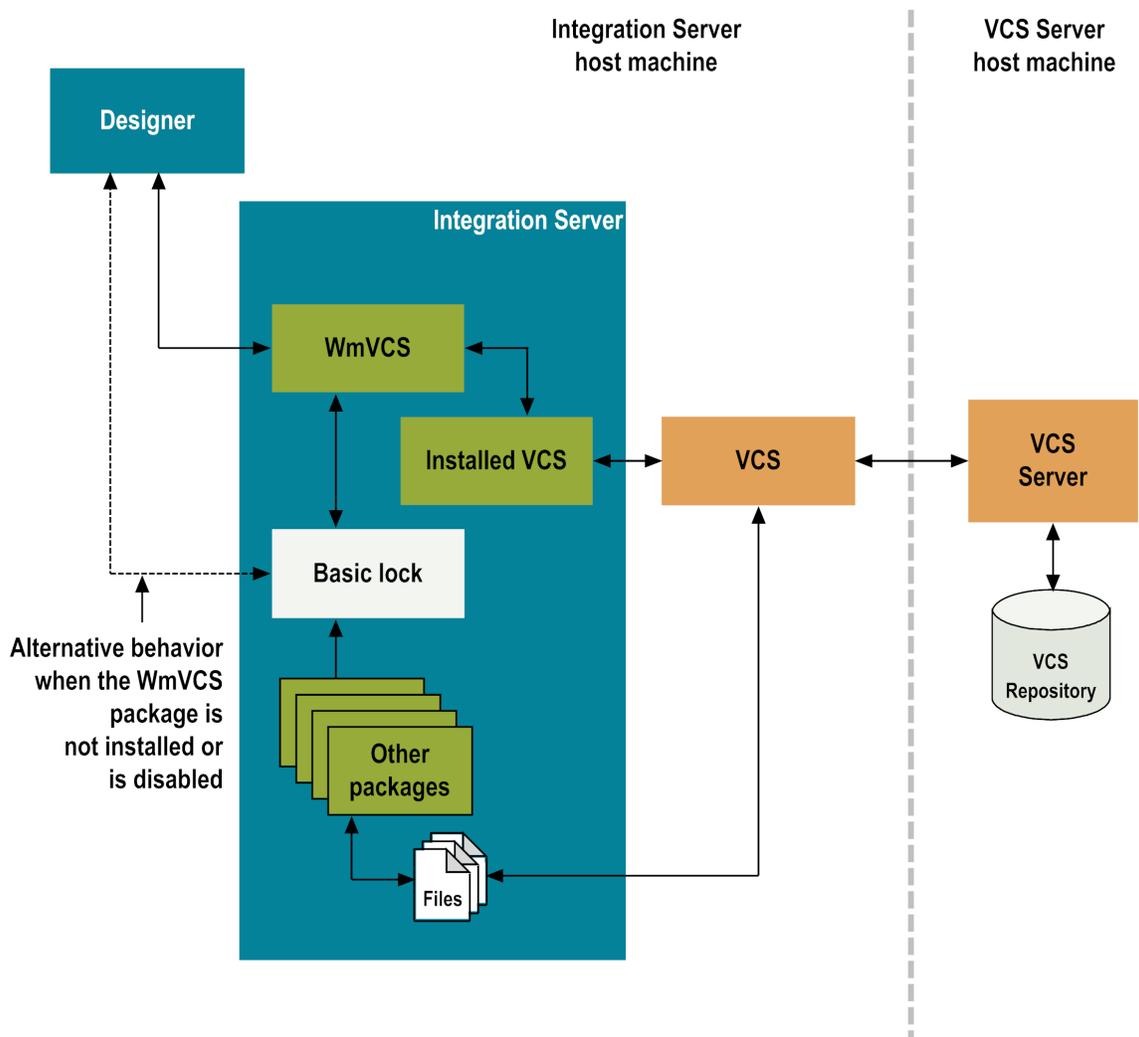
Terminology

As an integrated development environment (IDE), Designer can be used to create business integration solutions. Within the architecture of Integration Server, each item in the Package Navigator view of Designer is referred to as an *element*. The term *element* also applies to packages and folders, but because the VCS Integration feature treats packages and folders slightly differently than other elements, the terms *package* and *folder* are used explicitly throughout this guide. The term *element* is used to refer generally to services and other items contained within a folder in a package, such as document types, schemas, triggers, rule services, and web service descriptors.

Important: The VCS Integration feature considers a *folder* an organizational container. Although VCS commands can be applied to a folder (effectively applying the command to the contents of the folder), folders themselves are not displayed as checked out.

Architecture

The VCS Integration feature is installed on and operates within Integration Server, although its functionality is made available through the Designer user interface. The following simplified diagram describes the interaction of the VCS Integration feature with Integration Server and Designer:



Basic Concepts

The VCS Integration feature consists of the WmVCS package and a second package that is specific to the VCS you are using, as follows:

This package...	For this VCS...
WmSourceSafe	Visual SourceSafe
WmClearCase	ClearCase
WmSubversion	Subversion

You can connect to one VCS repository from each Integration Server.

Designer continue to be fully compatible with Integration Servers that do not have the VCS Integration feature installed; when Designer connects to an Integration Server where the VCS Integration feature is not installed, is disabled, or otherwise does not start, the basic locking functionality remains available in Designer.

To use the VCS Integration feature to check packages and elements in and out of your VCS repository from Designer, the VCS client software must be installed on the computer on which Integration Server is running. Additionally, the VCS client software must be executable by the user account under which Integration Server is running. You must also have Access Control List (ACL) write privileges on the Integration Server for the packages, folders, and elements with which you want to work.

The VCS Integration feature does not provide you with a direct connection to your VCS. If you cannot connect to the VCS through your VCS client software, you will receive an error message each time you attempt to check packages, folders, or elements in to or out of the VCS from Designer.

When you start Designer and connect to an Integration Server that has the VCS Integration feature installed, configured, and enabled, the **File** menu and the pop-up menu are modified to offer various VCS-specific commands. In addition, the existing **Delete** command (**Edit > Delete** on the main menu bar, and **Delete** on the pop-up menu) interacts with the VCS Integration feature.

You can apply the VCS Integration menu commands to folders and elements that appear in the Designer Package Navigator view, up to the package level.

Components of your webMethods solutions that do not appear in the Navigation panel (such as dynamic server pages, HTML documents, templates, and configuration files) must be manually checked in and out of the VCS repository with your VCS client.

In addition, when you use Designer to apply copy and paste, move, and rename actions to any package, folder, or element subject to VCS control, the VCS Integration feature automatically updates the VCS repository to reflect the changes. For example, if you

copy a service from one folder to another, the copied service is automatically added to the VCS repository in the new location.

Similarly, if you move a folder or element, it will be removed from its original location in the VCS repository and added to the VCS repository in its new location. Renamed packages, folders, or elements are also automatically updated in the VCS repository.

Summary of VCS Commands

The following table provides a summary of VCS commands. For more detailed information, see *webMethods Service Development Help*.

Command	Description
Check Out	<p>Checks out the selected element. When this command is applied to a package, or to a folder in a package, all of the supported elements within the container's hierarchy are checked out. A check mark appears next to a checked out element. Packages and folders do not display the checked out icon.</p> <p>Note: If an element has not been previously added to the VCS, applying the Check Out command adds the element to the VCS and places it in a checked out state.</p>
Check In	<p>Checks in the selected element. When this command is applied to a package, or to a folder in a package, all of the checked out elements within the container's hierarchy are checked in. The check mark next to the element in the Navigation panel is removed.</p> <p>Note: If an element has not been previously added to the VCS, applying the Check In command adds the element to the VCS and places it in a checked in state.</p>
View History	<p>Displays the View History dialog box with a revision history of the folder or element. This command also displays the lock status of the selected folder or element. Not available for packages.</p>
Revert Changes	<p>Discards all changes made to an element checked out by the current user. When this command is applied to a package, or to a folder in a package, changes are discarded for all of the checked out elements within the container's hierarchy. The Integration Server version of the element is updated with the last version committed to the VCS, and the check mark next</p>

Command	Description
	to the file in the Navigation panel is removed. After the revert procedure, the entire package is reloaded in Designer.
Get Earlier Version	<p>Updates the Integration Server version of the selected element with an earlier version in the VCS, specified by revision date or VCS label. When this command is applied to a package or to a folder in a package, all supported elements within the container's hierarchy are updated. After the update procedure, the entire package is reloaded in Designer.</p> <p>If a package or folder contains checked out elements, this command is unavailable.</p>
Get Latest Version	<p>Updates the Integration Server version of the selected element with the most current version in the VCS. When this command is applied to a package or to a folder in a package, all supported elements within the container's hierarchy are updated. After the update procedure, the entire package is reloaded in Designer.</p> <p>If a package or folder contains checked out elements, this command is unavailable.</p>
Delete	<p>Deletes the selected package, folder, or element from Integration Server and the VCS. When this command is applied to a package, or to a folder in a package, all supported elements within the container's hierarchy are deleted from Integration Server and the VCS.</p> <p>If a package contains checked out elements, Designer shows an error.</p>

Hierarchical Operation

The VCS Integration feature works with IS elements up to and including the package level. When you apply a VCS Integration command to a container element (a folder or package), the command is applied to all of the folders and elements in the container that are subject to VCS management. You can use this feature to check out or check in elements, or get the latest version or an earlier version of all the elements in an entire package or folder

For example, if you apply the **Check In** command to a folder, all supported elements contained in the hierarchy of that folder are checked in to the VCS. For specific information about files that are not supported by the VCS Integration feature, see [“Understanding Package, Element, and File Behavior” on page 19](#).

Note: When you work with a package with many elements, it may take a significant amount of time to check the package in or out; Designer will not be available during the check in or check out procedure.

Typical Work Flow

To use the VCS Integration feature to manage your webMethods packages, folders, and elements, the typical work flow is to:

1. Select the package, folder, or element on the Integration Server instance that you want to place in your VCS.
2. Check in the selected item to your VCS repository.
3. Check out the package, folder, or element for modification.

Understanding Package, Element, and File Behavior

The VCS Integration feature works with all of the packages and IS elements displayed in the Navigation panel or the Package Navigator view.

The elements in a package are in turn supported by various files maintained within the Integration Server instance directory structure. However, Designer modify the content of these files based on the work you do in Designer. You do not work with these files directly.

Note: You must have Access Control List (ACL) write privileges on the Integration Server for the packages, folders, and elements you want to work with.

Prior to the release of the VCS Integration feature, users who wanted to maintain these files in a version control system had to locate them in the Integration Server instance directory structure and manually check them in to and out of the VCS repository. The VCS Integration feature enables you to check in and check out packages and elements directly from within Designer; the underlying files are automatically checked in and out of the VCS repository for you.

Package files that do not appear in the Navigation panel are not subject to the VCS Integration feature. For a full description of supported and unsupported package files, see [“Package Contents and Interaction with the VCS Integration Feature”](#) on page 20, below.

Location of a Package

A package is represented by a single directory within the Integration Server instance. This directory has the same name as the package it represents and resides in the *Integration Server_directory*\instances*instance_name* \packages*package* directory.

Package Contents and Interaction with the VCS Integration Feature

The following sections describe the directories that contain files that support the elements contained in a package. The sections also indicate whether the underlying files are subject to the VCS Integration feature.

..\packages\packageName Directory

This directory contains a manifest.v3 file with information about the package, including its status, version information, dependencies on other packages, and the names of its startup and shutdown services.

When a new package is created in Designer, the manifest.v3 file is checked in to the VCS repository along with the package. Thereafter, the VCS Integration feature ignores the manifest.v3. For more information, see [“About the Manifest File” on page 52](#).

\code Directory

This directory does not normally contain files, only subdirectories that hold the source and executable code for coded services (that is, Java, C/C++, and COM services).

Subdirectory	Contents	VCS Integration
\code\jars	External Java libraries used by the services in the package.	You must manually check in and check out files in this directory using your VCS client.
\code\libs	All executable C/C++ and/or COM program files that belong to the package. These files can be one of the following: <ul style="list-style-type: none"> ■ Third-party C/C++ or COM libraries used by services in this package ■ Libraries and/or executable files for services that you have developed using C/C++ and/or COM 	You must manually check in and check out files in this directory using your VCS client.
\code\classes	Java class files for services that belong to the package.	You must manually check in and check out files in this directory using your VCS client. Generated code is not usually version controlled.

Subdirectory	Contents	VCS Integration
\code \source	<p>Source files for coded services (Java services, C/C++ services, and COM services) that belong to the package.</p> <p>If the package contains Java services, this directory will have one source file for each folder that contains Java services. This file holds the source code (as a single class) for all of the Java services (coded as methods of the class) in the folder.</p>	<p>Some files in this directory are automatically checked in and checked out by the VCS Integration feature:</p> <ul style="list-style-type: none"> ■ Files for Java services are checked in and out by the VCS Integration feature. ■ C/C++ or COM services are added to the VCS repository when they are created; after that, they must be manually checked in and checked out.

\config Directory

This directory contains configuration information for the package and its services, such as the properties.cnf file. You must manually check in and check out files in this directory using your VCS client.

\doc Directory

This directory contains any user-created documentation for the package, in PDF or HTML format. You must manually check in and check out files in this directory using your VCS client.

\lib Directory

This directory has been replaced by the \resources directory, but still exists for backwards-compatibility purposes. You must manually check in and check out files in this directory using your VCS client.

\ns Directory

This directory contains subdirectories for each top-level folder that resides in the package.

\ns Subdirectory	Contents	VCS Integration
\FolderName	The file node.idf, and possibly subdirectories for subfolders, as well as elements such as services, schemas, and triggers.	Files in this directory are automatically checked in and checked out by the VCS Integration feature.

\Ins Subdirectory	Contents	VCS Integration
		<p>If the folder contains Java services, the node.idf file holds Java code that is shared among the Java services in the folder.</p> <ul style="list-style-type: none"> ■ If you maintain Java services using Designer, these files are updated automatically. ■ If you maintain Java services using your own IDE, you must run the jcode frag command on the folder's Java source file before you check in the package.

\FolderName \
FlowServiceName

These files:

File	Description	VCS Integration
node.ndf	Service's input/output parameters and run-time setting	Files in this directory are automatically checked in and checked out by the VCS Integration feature.
flow.xml	Service's logic	<p>Note: Backup files generated by Designer or Integration Server are not checked in and checked out by the VCS Integration feature.</p>
XsltServiceName .xsl	XSLT style sheet, created only for XSLT services	

\FolderName \
CodedServiceName

These files for coded (Java, C/C++, or COM) services:

\Ins Subdirectory	Contents		VCS Integration
	File	Description	VCS Integration
	node.ndf	Service's input/output parameters and run-time settings	The node.ndf file is updated when you edit the service's run-time settings using Integration Server Administrator or Designer.
	java.frag	Compiled code for the service	<p>The java.frag file is updated when you modify and save the source code for the Java service with Designer.</p> <p>If you maintain Java services with your own IDE, the java.frag file is replaced when you run the jcode frag command on the Java source file for this folder.</p>
<i>\FolderName \ non-ServiceElementName</i>	<p>The file node.ndf, which holds the definition of the element.</p> <p>The non-service elements include specifications, document types, triggers, schemas, flat file schemas, flat file dictionaries, and adapter notifications.</p>		Files in this directory are automatically checked in and checked out by the VCS Integration feature.
<i>\RuleProject \RuleInstance</i>	These files:		
	File	Description	VCS Integration
	node.ndf	Service's input/output parameters and run-	Files in this directory are automatically checked in and checked out by the VCS Integration feature.

Ins Subdirectory	Contents	VCS Integration
	<div style="text-align: right; margin-bottom: 10px;">time settings</div> <hr style="width: 50%; margin: 0 auto;"/> flow.xml Service's logic	Note: Backup files generated by Designer or Integration Server are not checked in and checked out by the VCS Integration feature.

pub Directory

This directory contains HTML pages, XML input files, DSPs, and related image files that are served to HTTP clients (this is the WWW root directory for the package). You must manually check in and check out files in this directory using your VCS client.

resources Directory

This directory contains Java classpath resources, which are external resources (such as zip files or a properties file) that can be loaded by specifying a relative path, resolved relative to the classpath, instead of an absolute path. This makes the code more portable. You must manually check in and check out files in this directory using your VCS client.

templates Directory

This directory contains output templates that have been created for the services in this package. You must manually check in and check out files in this directory using your VCS client.

web Directory

This directory contains user-created JavaServer Pages (JSPs), servlets, and associated files used for web applications. These files may be stored individually or packaged in a single web archive (.war) file. For more information, see the *Web Applications Developer's Guide*. You must manually check in and check out files in this directory using your VCS client.

Applying Labels within the VCS Repository

The VCS Integration feature enables you to get an earlier version of VCS files by specifying a label. VCS labels must be created with the VCS client; the VCS Integration feature does not enable you to create VCS labels from within Designer.

When applying a version label to files in the VCS repository, it is recommended that you apply the version label at the package level, thereby including all folders and elements within the package hierarchy.

Working with Blaze Rule Services

When you deploy a rule from Blaze Advisor, a new package, a folder, a subfolder and a rule node are created in Integration Server. However, only the package is checked in to the VCS repository, during the rule deployment. You must check in or check out the folders, subfolders, and rule node manually.

Working with Web Service Descriptors

When a web service descriptor is created in Designer, a web service descriptor node and associated schemas and document types are created in Integration Server. However, only the web service descriptor node is checked in to the VCS repository. You must check in or check out the associated schemas and document types manually.

2 System Requirements

■ Platform, Hardware, and VCS Requirements	28
■ Visual SourceSafe Configuration	29
■ ClearCase Configuration	30
■ Subversion Configuration	30

Platform, Hardware, and VCS Requirements

The VCS Integration feature has no operating system, JDK, or hardware requirements beyond those of the host Integration Server and the VCS client.

The VCS Integration feature requires Integration Server version 8.0 or higher and Designer 8.0 or higher.

After the initial installation, Designer can connect to supported Integration Servers that are running with or without the VCS Integration feature.

The VCS Integration package is designed to work with the following version control systems (VCSes):

VCS	Version
Microsoft Visual SourceSafe	6.0 and 2005
IBM Rational ClearCase	7.0
Subversion	1.4, 1.6, and 1.7

The following are required for Integration Server to work with any VCS:

- The client software for the VCS must be installed on the same computer as the Integration Server that is running the VCS Integration feature.
- The VCS client software must be executable by the user account that is used to start the Integration Server, and must be included in the PATH statement for that user account.
- Each Integration Server user that interacts with the VCS Integration feature must have full read and write permission for the files the user is checking into and out of the VCS repository.

The VCS Integration feature does not provide you with a direct connection to your VCS. If you cannot connect to the VCS through your VCS client software, you will receive an error message each time you try to check packages, folders, or elements in to or out of the VCS from Designer.

The VCS Integration feature offers no support for configuring or manipulating the VCS server environment.

Visual SourceSafe Configuration

Complete the following procedure for each Integration Server user that will use Visual SourceSafe (that is, for whom you will create a mapping, as described in [“Configuring Access to the VCS Repository”](#) on page 39).

To configure Visual SourceSafe for an Integration Server user

1. Identify the Visual SourceSafe database to the user's Windows system, as follows:
 - a. Right click **My Computer** and then click **Properties**.
 - b. Click the **Advanced** tab and then click **Environment Variables**.
 - c. In the **System variables** table, do the following:
 - a. To the Path variable, add the fully qualified path to the Visual SourceSafe folder that contains the binary files for commands and the ssexp.exe file.
 - b. Add the variable SSDIR and set it to the fully qualified path to the Visual SourceSafe database.
 - d. Click **OK** three times.

Note: These instructions are for Windows XP Professional. Instructions for other Windows systems might vary slightly.

2. Set Visual SourceSafe client options as follows:
 - a. Start the Visual SourceSafe client.
 - b. Click **Tools**, then click **Options**.
 - c. Click the **Local Files** tab, clear the **Append end-of-line to all text files** check box, and click **OK**.
 - d. It is recommended that you configure Visual SourceSafe to check in unchanged files. Doing so ensures that check in comments are recorded for all elements, and that all related files that have been checked out with a package, folder, or element will have their VCS version numbers incremented simultaneously. Click the **General** tab, select **Check in** from the **Check in unchanged files** list, and click **OK**.
 - e. If Visual SourceSafe is configured to use a text editor such as Notepad for viewing files, the text editor will open each time the VCS Integration feature checks in a file to the Visual SourceSafe repository. To avoid this, remove the text editor specification. Click the **General** tab, delete the text in the **Editor for viewing files** box, and click **OK**.

ClearCase Configuration

Complete the following procedure for each Integration Server user that will use ClearCase.

To configure ClearCase for use with an Integration Server user

1. If the user is on a UNIX system, identify the ClearCase command binaries to the system. To the PATH variable, add the fully qualified path to the ClearCase folder that contains the binary files for commands.
2. If the user is on a Windows system identify the ClearCase command binaries to the system as follows:
 - a. Right click **My Computer** and then click **Properties**.
 - b. Click the **Advanced** tab and then click **Environment Variables**.
 - c. In the **System Variables** table, to the Path variable, add the fully qualified path to the ClearCase folder that contains the binary files for commands.

Note: These files are usually present by default. However, if they are not present, you must add them.

- d. Click **OK** three times.

Note: These instructions are for Windows XP Professional. Instructions for other Windows systems might vary slightly.

3. If the user is on a Windows system configure ClearCase Dynamic Views to work with the VCS Integration feature as follows:
 - a. On the **Start** menu, go to **Settings > Control Panel > ClearCase**.
 - b. Click the **MVFS** tab.
 - c. Select the check boxes for **Case Insensitive MVFS** and **Case Preserving**.
 - d. Click **OK**.

Subversion Configuration

Complete the following procedure to configure each Integration Server user to work with Subversion.

To configure Subversion for use with an Integration Server user

1. In the Subversion server, create a repository for Integration Server to use.

2. Share the Subversion root directory in the network, which contains the repositories. Grant all Integration Server users read/write permission.
3. Go to the new *Subversion_repository* \conf directory and do the following:
 - a. Open the `svnserver.conf` file in a text editor, uncomment the line `password-db = passwd`, and save and close the file.
 - b. Open the `passwd` file in a text editor, uncomment or add the names of Integration Server users, and save and close the file.
4. Start the Subversion repository using the `svnserve` command.

Updating the Path Environment Variable For Subversion

For each Integration Server user that will use Subversion, follow the steps below to update the user's Path environment variable with the folder that contains the binary files for the Subversion client.

To update user's Path environment variables

- If the Integration Server is on a UNIX system, run the following command:

```
export PATH = $PATH:Subversion_directory
```

By default, the *Subversion_directory* is `/opt/CollabNet_Subversion/bin`.

- If the Integration Server is on a Windows system, do the following:
 1. Right-click **My Computer** and then click **Properties**.
 2. Click the **Advanced** tab and then click **Environment Variables**.
 3. In the **System variables** table, do the following:
 - a. In the **Variable** column, click **Path** and then click **Edit**.
 - b. Add the full path to the Subversion folder that contains the binary files for commands. By default, the full path is `C:\Program Files\Subversion\bin`.
 - c. Click **OK**.
 4. Click **OK** twice more.

Excluding File Types

During package publication, Integration Server includes all files associated with the items being published. In some cases, this includes files that are not required for the package to function. For example, when the publishing Integration Server uses Subversion to maintain files, the package contains read-only `.svn` files. If you are publishing the package as a full or partial replacement for a package of the same name on another Integration Server and that Integration Server also uses Subversion, then

the read-only .svn files from the publishing Integration Server do not replace the read-only .svn files on the subscribing server, and the publishing action will fail.

Integration Server provides the `watt.server.createPackage.ignorePattern` configuration parameter with which you can specify the types of files you do not want Integration Server to include when publishing packages.

For more information about `watt.server.createPackage.ignorePattern`, see *webMethods Integration Server Administrator's Guide*.

3

Configuring the VCS Integration Feature

■ Overview	34
■ Configuring Integration Server and Designer	34
■ Configuring the WmVCS Package to Work with Visual SourceSafe	35
■ Configuring the WmVCS Package to Work with ClearCase	36
■ Configuring the WmVCS Package to Work with Subversion	37
■ Configuring Access to the VCS Repository	39

Overview

You must configure webMethods products to operate with the VCS Integration feature and the VCS server. For Visual SourceSafe or Subversion, you must also map Integration Server users to Visual SourceSafe or Subversion users, respectively.

If you have not already done so, configure your VCS to conform to the requirements described in [“System Requirements” on page 27](#).

Configuring Integration Server and Designer

After you create your VCS administrator user account, you configure Integration Server and Designer, or both to work with the VCS Integration feature.

Removing the `watt.server.ns.backupNodes` Setting

If you have enabled the `watt.server.ns.backupNodes` setting, you must remove it. This setting conflicts with the operation of the VCS Integration feature, which essentially replaces it.

To remove the `watt.server.ns.backupNodes` setting

1. In the Integration Server Administrator, go to the **Settings > Extended** page.
2. Determine whether the `watt.server.ns.backupNodes` setting is displayed in the **Extended Settings** list. If it is, go to step 4. If the setting is not visible:
 - a. Click **Show and Hide Keys**.
 - b. On the **Settings > Extended > Show and Hide Keys** page, look for the setting `watt.server.ns.backupNodes`. If the setting is present, select the check box next to `watt.server.ns.backupNodes` and click **Save Changes**. If the setting is not visible, click **Return to Extended Settings**.
3. On the **Settings > Extended** page, click **Edit Extended Settings**.
4. In the **Extended Settings** text box, delete the `watt.server.ns.backupNodes` setting.
5. Click **Save Changes**.

Configuring Designer

If you have enabled the option in Designer to automatically unlock files upon saving, you must disable it. This option conflicts with the operation of the VCS Integration feature.

To disable the automatic unlock upon save option in Designer

1. Start Designer.
2. Go to **Windows > Preferences > Software AG > Service Development > Package Navigator**.
3. Clear the **Automatically unlock upon save** check box, and then click **OK**.

Configuring the WmVCS Package to Work with Visual SourceSafe

1. In Integration Server Administrator, go to the **Solutions > VCS > Edit Configuration** page.
2. In the **Select Version Control System** list, click **Microsoft Visual SourceSafe**, and then click **Save Changes**.
3. Restart Integration Server.
4. Go to the **Solutions > VCS > Edit Configuration** page, click the **Advanced Settings** link, and complete the fields as follows:

Field	Entry
Command Timeout (msec)	<p>Period of time, in milliseconds, that a command has in which to execute. If the command does not execute within that period, it times out.</p> <p>If you want to use the default of 60000 (one minute), leave this field blank. To change the default, specify a positive integer. If you want to have no timeout, enter -1.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: The timeout is applied to each command. When Visual SourceSafe operations join several commands, the timeout is the sum of the timeouts for all the commands.</p> </div>
Working Folder	<p>Fully qualified path to the Visual SourceSafe working folder in the Integration Server file system in which to store your working files. Software AG recommends that you specify the <i>Integration Server_directory/instances/instance_name/packages</i> directory, or an individual package in that directory.</p>
VSS Project Name	<p>Name of the project in Visual SourceSafe that contains all of the files and directories you work on. Typically, this path begins with a dollar sign (\$), to indicate the Visual</p>

Field	Entry
	SourceSafe root project (for example: \$/Project/Team/Module).
Allow Multiple Checkouts	If Visual SourceSafe is configured for exclusive check out, clear this check box. If Visual SourceSafe is configured for non-exclusive check out, select this check box.
	<p>Important: Software AG recommends that you configure your VCS for exclusive locking to prevent users from checking out files from the VCS client after they have been checked out from Designer. The VCS Integration feature does not support "diff" or merge operations. If you use non-exclusive checkout and multiple users check out the same element, only the first user to check in will be successful. The other users will receive error messages.</p>

5. Click **Save Changes**.
6. Restart Integration Server.

Configuring the WmVCS Package to Work with ClearCase

1. In Integration Server Administrator, go to the **Solutions > VCS > Edit Configuration** page.
2. In the **Select Version Control System** list, click **ClearCase**, and then click **Save Changes**.
3. Restart Integration Server.
4. Go to the **Solutions > VCS > Edit Configuration** page, click the **Advanced Settings** link, and complete the fields as follows:

Field	Entry
Command Timeout (msec)	<p>Period of time, in milliseconds, that a command has in which to execute. If the command does not execute within that period, it times out.</p> <p>If you want to use the default of 60000 (one minute), leave this field blank. To change the default, specify a positive integer. If you want to have no timeout, enter -1.</p> <p>Note: The timeout is applied to each command. When ClearCase operations join several commands, the timeout is the sum of the timeouts for all the commands.</p>

Field	Entry
ClearCase View Directory	<p>By default, Integration Server assumes that the files you work on are in a folder that is part of a Dynamic or SnapShot view. If you are going to specify a working folder, identify the mapping folder in a ClearCase Dynamic or Snapshot view.</p> <p>Note: If the Integration Server is installed in a Dynamic view, check the <i>Integration Server_directory</i>\instances\<i>instance_name</i> \packages folder into ClearCase before working with VCS Integration feature.</p>
Working Folder	<p>If you work on files that reside in a folder that is not part of a Dynamic or Snapshot view, specify the name of the folder. Software AG recommends that you specify the <i>Integration Server_directory</i>\instances\<i>instance_name</i> \packages directory, or an individual package in that directory.</p>
Checkout Mode	<p>ClearCase lets you choose whether your checkouts are Reserved, Unreserved, or Both. If you choose Unreserved, make sure your ClearCase VOB and View settings do not restrict unreserved check outs.</p> <p>Important: The VCS Integration feature does not support "diff" or merge operations. If you set the checkout mode to Unreserved and multiple users check out the same element, only the first user to check in will be successful. The other users will receive error messages.</p>
ClearCase Branch Name	<p>This field determines the branch used for the Get earlier version command. By default, ClearCase assumes that files you want to use reside on the main ClearCase branch. If they do not, identify the branch on which they do reside.</p>

5. Click **Save Changes**.
6. Restart Integration Server.

Configuring the WmVCS Package to Work with Subversion

1. In Integration Server Administrator, go to the **Solutions > VCS > Edit Configuration** page.
2. In the **Select Version Control System** list, click **Subversion**, and then click **Save Changes**.
3. Restart Integration Server.

Connecting to Subversion for the First Time

To connect to Subversion for the first time

1. In Integration Server Administrator, go to the **Solutions > VCS > Edit Configuration** page and click the **Advanced Settings** link.
2. In the **Subversion Repository Location** field, type the URL for the Subversion repository.
3. Click **Save Changes**.
4. Restart Integration Server.

Disconnecting from Subversion

You might need to disconnect Integration Server from the Subversion repository at some point. The IP address for the Subversion repository might change, for example, or you might want to connect to a different Subversion repository, or you might want to use a different source control system altogether.

To disconnect from the Subversion repository

1. Go to the **Solutions > VCS > Edit Configuration** page and click the **Advanced Settings** link.
2. Subversion creates metadata files on Integration Server. If you no longer need the files (for example, because you are going to use a different source control system), select the **Delete Subversion Metadata Files** check box.
3. Click **Disconnect**.

Reconnecting to Subversion

If you are still using Subversion as your source control system, you need to reconnect.

To reconnect to the Subversion repository

1. Go to the **Solutions > VCS > Edit Configuration** page and click the **Advanced Settings** link.
2. In the **Subversion Repository Location** field, type the URL for the Subversion repository.
3. If you are reconnecting to the same Subversion repository at a different IP address, select the **Reconnect to Subversion** check box.
4. Click **Save Changes**.
5. Restart Integration Server.

Configuring Integration Server to Connect to a Subversion Repository through a Non-English Subversion Client

When the VCS Integration feature on an Integration Server localized to English is configured to work with a Subversion client in a locale other than English, unexpected behavior might occur during check in and check out operations. This behavior occurs because the request and response from an Integration Server running in an English locale is misinterpreted by the Subversion client running on a non-English platform. To prevent this, you need to make a change to the platform environment.

To configure the platform environment when using a non-English Subversion client with Integration Server

- For Windows, add the following to System environment variable

```
LC_MESSAGES=en_EN
```

- For UNIX/Linux, add the following to profile

```
export LC_MESSAGES=en_EN
```

Note: After making the above change, the `svn --version` command client should always result in a response in the English language.

Configuring Access to the VCS Repository

Important: The feature described in this section is required for Visual SourceSafe and Subversion.

In a shared development environment, a software developer might have one user name on Integration Server and a different user name on the VCS server. The VCS Integration feature accommodates this by enabling you to map an Integration Server user to a VCS user.

Integration Server users gain access to the VCS server by being mapped to a VCS user name and password. If you make changes to the VCS user name or password, you must edit the mapping to reflect the changes. For example, if you change the VCS user password, you must change the VCS user password in the mapping.

Each Integration Server user can map to only one VCS user. If you want to access the VCS server using different VCS users at different times, you can edit your mapping each time you want to use a different VCS user, or you can set up multiple Integration Server users and map each one to a different VCS user.

When you check in or delete a file, the VCS Integration feature records both the Integration Server user and the VCS user as part of the VCS revision history. For more information on working with multiple user accounts in a shared development

environment, see [“Understanding Multiple User Accounts and VCS Integration”](#) on page 53.

If you are an Integration Server user with administrator privileges, you can also create mappings for other Integration Server users.

Note: Only map multiple Integration Server users to one VCS user if this behavior is required in your environment. Depending on your VCS, mapping multiple Integration Server users to a single VCS user might result in errors or unpredictable results. In addition, you might have difficulty tracing a VCS action back to a particular Integration Server user.

Working with a Windows System

When the VCS client is on a Windows system, the system provides the credentials of the currently logged on Windows user by default when VCS commands are submitted if no other user credentials are defined. Therefore, if the user credentials for the Windows system are the same as those of a VCS user, that Windows user will be able to check files in to and out of the VCS server even if a VCS user has not been defined for that user.

For example, suppose you log on to Windows as user Pamela with a password of 1234X. You then log on to Designer as user Dev01 with a password of ABCD1. If you have a VCS user Pamela with a password of 1234X, you will be able to access the VCS repository even though your Integration Server user is not mapped to your VCS user. However, VCS Integration feature will have no Integration Server user to record in the VCS revision history transaction and will record only the VCS user.

Creating VCS Users

You must create all VCS users on the VCS server, using the VCS server user management feature. The VCS Integration feature provides no means of managing user accounts on the VCS server. The VCS users must have read and write permission for the package and element files they will be working with. For security considerations, limit each VCS user to the minimum file permissions and system rights required to carry out work on the packages and elements that the user will be working with.

Mapping One Integration Server User to One VCS User

To map one Integration Server user to one VCS user

1. In the Integration Server Administrator, go to the **Solutions > VCS > User Mapping > Create New User Mapping** page.
2. In the **Integration Server User Name** field, type the Integration Server user name and the VCS user name and password. Use the format *Integration Server user name ;VCS user name ;VCS password* . Type the entry for each Integration Server user on its own line. All user account credentials are case-sensitive.

Important: If you specify an invalid VCS user name and password, when commands are sent to the VCS client, the client will stop responding while awaiting the correct user name and password. As a result, Designer will also stop responding. Integration Server Administrator does not validate, create, or modify VCS users.

3. Click **Save Changes**.

The VCS Integration feature encrypts the VCS passwords and stores the VCS user names and encrypted passwords on Integration Server.

Mapping Multiple Integration Server Users to One VCS User

Important: Only map multiple Integration Server users to one VCS user if this behavior is required in your environment. Depending on your VCS, mapping multiple Integration Server users to a single VCS user might result in errors or unpredictable results. In addition, you might have difficulty tracing a VCS action back to a particular Integration Server user.

To map multiple Integration Server users to one VCS user

1. In the Integration Server Administrator, go to the **Solutions > VCS > User Mapping > Create New User Mapping** page.
2. In the **Integration Server User Name** field, type the Integration Server user names, one on each line.
3. In the **VCS User Name**, **VCS Password**, and **Confirm VCS Password** fields, type the VCS user name and password. All user credentials are case-sensitive.

Important: If you specify an invalid VCS user name and password, when commands are sent to the VCS client, the client will stop responding while awaiting the correct user name and password. As a result, Designer will also stop responding. Integration Server Administrator does not validate, create, or modify VCS users.

4. Click **Save Changes**.

The VCS Integration feature encrypts the VCS passwords and stores the VCS user names and encrypted passwords on Integration Server.

Editing a User Mapping

To edit a user mapping

1. In Designer, check in all files for the Integration Server user whose mapping you want to edit.

Important: VCS will not permit a file to be checked in by a user other than the user who checked it out. If you do not check in the files, the user's packages, folders, and elements will remain checked out and unavailable to other VCS users.

2. In the Integration Server Administrator, go to the **Solutions > VCS > User Mapping** page.
3. In the Edit column for the user whose mapping to edit, click the **Edit** link.
4. In the **VCS User Name**, **VCS Password**, and **Confirm VCS Password** fields, type the VCS user name and password to map to the user.

Important: If you specify an invalid VCS user name and password, when commands are sent to the VCS client, the client will stop responding while awaiting the correct user name and password. As a result, Designer will also stop responding. Integration Server Administrator does not validate, create, or modify VCS users.

5. Click **Update Changes**.

The VCS Integration feature encrypts the VCS passwords and stores the VCS user names and encrypted passwords on Integration Server.

Deleting a User Mapping

In some cases, after you delete the mapping for an Integration Server user, the user might still have access to the VCS repository. For more information, see [“Working with a Windows System” on page 40](#).

To delete a user mapping

1. In Designer, check in all files for the Integration Server user whose mapping you want to delete.

Important: VCS will not permit a file to be checked in by a user other than the user who checked it out. If you do not check in the files, the user's packages, folders, and elements will remain checked out and unavailable to other VCS users.

2. In the Integration Server Administrator, go to the **Solutions > VCS > User Mapping** page.
3. In the Delete column for the user whose mapping to edit, click **X**.

Enabling and Disabling the VCS Integration Packages

After installation and configuration, the VCS Integration packages are enabled automatically each time the Integration Server is started. However, if you want to temporarily turn off the VCS Integration feature, you can disable the packages.

When the VCS Integration packages are disabled, only the basic locking functionality is available in Designer. For more information on the details of basic locking operation, see *webMethods Integration Server Administrator's Guide*.

Important: Any changes you make to packages and elements that exist in the VCS will not be applied to the VCS versions of those elements while the VCS Integration feature is disabled. If you plan to modify elements that exist in the VCS while the VCS Integration feature is temporarily disabled, check out those elements before disabling the VCS Integration packages. The checked out elements will not be available to other users on the VCS server as long as you have them checked out.

Temporarily Disabling the VCS Integration Feature

Before you disable the VCS Integration feature, check out any elements that you may want to work on while the VCS Integration feature is disabled. Elements that you do not check out will be recognized as "system locked" (that is, they remain locked by the VCS).

If you want to display the system locked status while you have the VCS Integration feature disabled, set the locking mode to "full" as described in [“Permanently Disabling the VCS Integration Feature” on page 44](#). If you do not change the locking mode, you will still be unable to modify the system locked files, but the system locked icon will not appear in the Navigation panel.

To temporarily disable the VCS Integration feature

1. Open the Integration Server Administrator if it is not already open.
2. In the **Packages** menu of the navigation area, click **Management**.
3. Locate the WmVCS package in the **Package List**.
4. Click **Yes** in the **Enabled** column for the WmVCS package. A confirmation message appears; click **OK** to disable the package. After the package is disabled, the **Enabled** column value changes to **No**.
5. Locate the package for your specific VCS system (WmSourceSafe, WmClearCase, or WmSubversion).
6. Click **Yes** in the **Enabled** column for the VCS package. A confirmation message appears; click **OK** to disable the package.
After the package is disabled, the **Enabled** column value changes to **No**.
7. If Designer is running, close the session with Integration Server and then open a new session.

The disabled packages will remain disabled until you enable them using the Integration Server Administrator. In Designer, basic locking and unlocking is now enabled.

Note: The next time you start Integration Server, the package for your specific VCS system (for example, WmSourceSafe) will not be displayed in the list of

packages on the **Packages > Management** page. It has been placed in inactive status and can be restored by clicking **Activate Inactive Packages**.

Enabling the VCS Integration Feature

To enable the VCS Integration feature after it has been disabled

1. Open the Integration Server Administrator if it is not already open.
2. In the **Packages** menu of the navigation area, click **Management**.
3. Locate the WmVCS package in the **Package List**.
4. Click **No** in the **Enabled** column for the WmVCS package. A confirmation message appears; click **OK** to enable the package. After the package is enabled, the **Enabled** column value changes to **Yes** and a check mark appears in the **Enabled** column.
5. Click **Activate Inactive Packages**.
6. On the **Packages > Management > Activate Package** page, select the package for your specific VCS system (for example, WmSourceSafe) in the **Inactive Packages** list.
7. Click **Activate Package**.
8. Click **Return to Package Management**.
9. In the **Package List**, locate the package for your specific VCS system, for example, WmSourceSafe.
10. Click **No** in the **Enabled** column for the WmVCS package. A confirmation message appears; click **OK** to enable the package. After the package is enabled, the **Enabled** column value changes to **Yes** and a check mark appears in the **Enabled** column.

Note: If you modified the **Version Control System** to **None** on the **VCS > Configuration > Edit Configuration** page, set it back to ClearCase or Microsoft Visual SourceSafe as described in [“Configuring the WmVCS Package to Work with Visual SourceSafe” on page 35](#), and then restart Integration Server.

11. If Designer is running, close the session with Integration Server and then open a new session.
12. Check in any elements that you may have worked on while the VCS Integration feature was disabled.

Permanently Disabling the VCS Integration Feature

Use the following procedure if you no longer want to integrate with a third-party version control system.

To permanently disable the VCS Integration feature

1. Disable the VCS Integration feature as described in [“Temporarily Disabling the VCS Integration Feature”](#) on page 43.
2. In Integration Server Administrator, go to the **Solutions > VCS** page.
3. Click **Edit Configuration**.
4. For **Version Control System**, click **None**.
5. Click **Save Changes**, then restart Integration Server to make the changes take effect.

Removing the VCS Support Package

You can remove the VCS-specific package from your Integration Server using the procedure below.

To remove the VCS support package

1. Start the Software AG Uninstaller as described in *Installing Software AG Products*.
2. In the product selection panel, open the Integration Server tree and select the VCS-specific package to remove (VSS Support or ClearCase Support).
3. Click **Next** and follow the instructions to complete the removal.

You cannot remove the WmVCS package, as it is a part of the basic Integration Server installation.

4 Working in a Shared Development Environment

■ General Behavior	48
■ Implementing a Shared Development Environment	49
■ Understanding Multiple User Accounts and VCS Integration	53

General Behavior

Although there are several ways to implement a shared development environment supporting multiple users (see [“Implementing a Shared Development Environment” on page 49](#)), they all share the following expected behavior:

VCS command	Expected behavior
Check In	<p>You can add existing packages and elements in the Navigation panel to the VCS repository by applying the Check In command. Any new packages and elements are automatically added to the VCS repository when they are created.</p>
Check Out	<p>You can check out any of the checked in packages or elements from the VCS repository by applying the Check Out command. If you apply the Check Out command to a folder that contains both checked in and checked out elements, you will receive a message naming the elements that are already checked out.</p> <p>In environments involving multiple Integration Servers sharing a single VCS server, it is possible that an element or package may appear to be available for check out when it is already checked out by another user, preventing you from checking it out (see “Working with Multiple Integration Servers and One VCS Server” on page 50).</p> <p>In any case, apply the View History command to the package or element to determine who has the item checked out.</p>
Delete	<p>You can delete existing packages and elements from both the Integration Server and the VCS repository by applying the Delete command. If an element is checked out by another user, you will receive a message that the package or element cannot be deleted.</p> <p>In environments involving multiple Integration Servers sharing a single VCS server, it is possible that an element or package may appear to be available for deletion when it is already checked out by another user, preventing you from deleting it (see “Working with Multiple Integration Servers and One VCS Server” on page 50).</p> <p>In any case, apply the View History command to the package or element to determine who has the item checked out.</p>

You cannot move or rename packages, folders, or elements checked out by others.

Implementing a Shared Development Environment

The VCS Integration feature is designed to support work in a multi-user, shared development environment. Each installation of Integration Server is unique, and development implementations vary widely; however, the following general formats are often encountered:

- Multiple software developers working on a single Integration Server that is connected to a single VCS server.
- Multiple software developers working on two or more Integration Servers, each Integration Server connected to a different, dedicated VCS server.
- Multiple software developers working on two or more Integration Servers, each Integration Server connected to the same VCS server.

The primary implementation consideration is that each Integration Server can be connected to one and only one VCS server. For Visual SourceSafe and Subversion, the secondary consideration is that, on each Integration Server, each Integration Server user must be mapped to a user on the Visual SourceSafe or Subversion server, respectively.

Working with a Single Integration Server

In this environment, multiple software developers work on a single Integration Server that is connected to a single VCS server. Setting up an installation where a single Integration Server is used for development involves the following general steps:

1. Configure the Integration Server for use with the VCS Integration feature. For more information, see [“Configuring Integration Server and Designer ” on page 34](#).
2. Configure the VCS-specific package to connect to the VCS server. For more information, see [“Configuring the WmVCS Package to Work with Visual SourceSafe” on page 35](#).
3. For Visual SourceSafe and Subversion, map Integration Server users to Visual SourceSafe or Subversion users, respectively. For more information, see [“Configuring Access to the VCS Repository” on page 39](#).

In this scenario, each Designer user is able to check Integration Server packages and elements in to and out of the VCS server.

Special considerations apply if you modify any of the package's settings, properties, or enabled/disabled status. For more information, see [“About the Manifest File” on page 52](#).

Working with Multiple Integration Servers and VCS Servers

In this environment, multiple software developers work on two or more Integration Servers, and each Integration Server is connected to a dedicated VCS server. This is essentially an extension of the single Integration Server/single VCS server environment. Setting up an installation for this environment involves the following general steps:

1. Configure each Integration Server for use with the VCS Integration feature. For more information, see [“Configuring Integration Server and Designer ” on page 34](#).
2. Configure the VCS-specific package on each Integration Server to connect to its dedicated VCS server. For more information, see [“Configuring the WmVCS Package to Work with Visual SourceSafe” on page 35](#).
3. For Visual SourceSafe and Subversion, map Integration Server users to Visual SourceSafe or Subversion users, respectively. For more information, see [“Configuring Access to the VCS Repository” on page 39](#).

In this scenario, each Designer user is able to check Integration Server packages and elements in to and out of the VCS server connected to that Integration Server. As a Designer user, you can connect only to the VCS server dedicated to the Integration Server you are logged on to.

Special considerations apply if you modify any of the package's settings, properties, or enabled/disabled status. For more information, see [“About the Manifest File” on page 52](#).

Working with Multiple Integration Servers and One VCS Server

In this environment, multiple software developers work on two or more Integration Servers, and each Integration Server is connected to the same VCS server. Setting up an installation for this environment involves the following general steps:

1. Configure each Integration Server for use with the VCS Integration feature. For more information, see [“Configuring Integration Server and Designer ” on page 34](#).
2. Configure the VCS-specific package on each Integration Server to connect to the same VCS server. For more information, see [“Configuring the WmVCS Package to Work with Visual SourceSafe” on page 35](#).
3. For Visual SourceSafe and Subversion, map Integration Server users to Visual SourceSafe or Subversion users, respectively. For more information, see [“Configuring Access to the VCS Repository” on page 39](#).

In this scenario, each Designer user is able to check Integration Server packages and elements in to and out of the VCS server. All Designer users connect to the same VCS server. There is a possibility that a package or element may be shown as available for check out when it is actually checked out by a user on another Integration Server. For an example, see [“Working with Multiple Integration Servers and One VCS Server” on page 50](#).

Sharing Packages Among Integration Servers

Packages can be created and checked in from one Integration Server, and then checked out from another Integration Server (using the VCS client), thereby adding the package to the second server (the package must contain a manifest.v3 file or it will not be recognized as a package by the second Integration Server; for more information, see [“About the Manifest File” on page 52](#)).

Note: When working in your development environment, you are advised to share packages manually (as described in [“Importing a Package from Another Integration Server” on page 51](#)), rather than use the package replication method. When you install a package replicated from another server, the manifest.v3 file will not be added to your VCS repository.

Locking Files in a Shared Development Environment

A key consideration in a shared development environment is the locking behavior of the VCS Integration feature.

For example, Maria is logged on to one Integration Server, and Ashish is logged on to a different Integration Server. Both Integration Servers are connected to the same VCS server and both Maria and Ashish are responsible for development of the same webMethods package, which exists on both Integration Servers. If Maria checks out element FlowService1, it is marked as checked out on both Maria's Integration Server and the VCS server. Because the VCS Integration feature does not validate the state of the element with the VCS server until a VCS command is applied to it, Ashish's Integration Server knows nothing of this and shows FlowService1 as available. When Ashish attempts to check out FlowService1, he will receive a message stating that the file cannot be checked out. This behavior also applies to the **Delete** command, and to the move and rename actions.

The potential for locking conflicts within shared packages increases when the **Check Out** command is applied at the folder or package level. In this case, all of the supported elements in the package or folder are checked out (if they are not already checked out by someone else), thereby locking them from access by other users. To mitigate this situation, you are advised to check out only the files you want to modify, and avoid checking out packages or folders unless you are reasonably certain no other developers will need access to them.

If multiple users perform operations on an element concurrently, unexpected results might occur.

Importing a Package from Another Integration Server

In some situations, someone will develop a package on another Integration Server, and you want to work with that package on your Integration Server. If you have VCS access to the package files on the VCS server, you can import the package into your Integration Server.

To import a package from another Integration Server into your Integration Server

1. On your Integration Server, create a new package with the exact same name as the package you want to import.
2. Apply the **Get Latest Version** command to the new package.

This will import the package and all its elements into your Integration Server. For additional information about the manifest file, see [“About the Manifest File” on page 52](#).

About the Manifest File

Each package must contain a manifest.v3 file in the `..\packages\package` directory. This file is created automatically when you create a package in Designer; it defines various package properties to the Integration Server such as the package name, version, and whether it is enabled or disabled.

You can modify a package's properties by right-clicking the package name in the Navigation panel and clicking **Open**. The various fields of the **Settings** and **Startup/Shutdown/Replication Services** tabs, and of the Properties panel, represent the contents of the manifest.v3 file. In addition, the manifest.v3 file indicates whether the package is enabled or disabled, as set in the Integration Server Administrator.

Whenever you modify any of these package settings or properties, you are changing the manifest.v3 file. This means that after you make these changes, the local version of the manifest.v3 file no longer matches the version in the VCS repository.

You must use your VCS client to add or update the manifest.v3 file in the VCS repository with the most current local version in the `..\packages\package` directory of the Integration Server. The exact procedure for doing this varies depending on your VCS.

If you are sharing development of the package with developers who are working on a different Integration Server, you must also notify those developers that the manifest.v3 file has been updated, and that they must manually update the package's manifest.v3 file on their Integration Server using their VCS client. The manifest.v3 file will not be updated automatically the next time they check out the package.

Note: Normally, checking in a file to the VCS places the local copy of the file in a read-only state, preventing changes to the file, and this is initially true for the manifest.v3 file. However, both Integration Server and Designer override the read-only state of the manifest.v3 file and place it into a read-write state when changes are saved.

Behavior of the manifest.v3 File When Creating a New Package

When you create a new package within Designer, the manifest.v3 file is added to the VCS repository by the VCS Integration feature. Thereafter, the manifest.v3 file is not checked in to or out of the VCS repository by the VCS Integration feature. You must

manually update the VCS repository version of the manifest.v3 file using your VCS client.

If you install a package using the Integration Server replication feature, the manifest.v3 file will not be added to the VCS repository. In this case, you must obtain a copy of the manifest.v3 file and add it to the VCS repository manually using the VCS client. Because of this issue, you are advised to share packages manually within your development environment (as described in [“Importing a Package from Another Integration Server” on page 51](#)), rather than use the package replication method.

Understanding Multiple User Accounts and VCS Integration

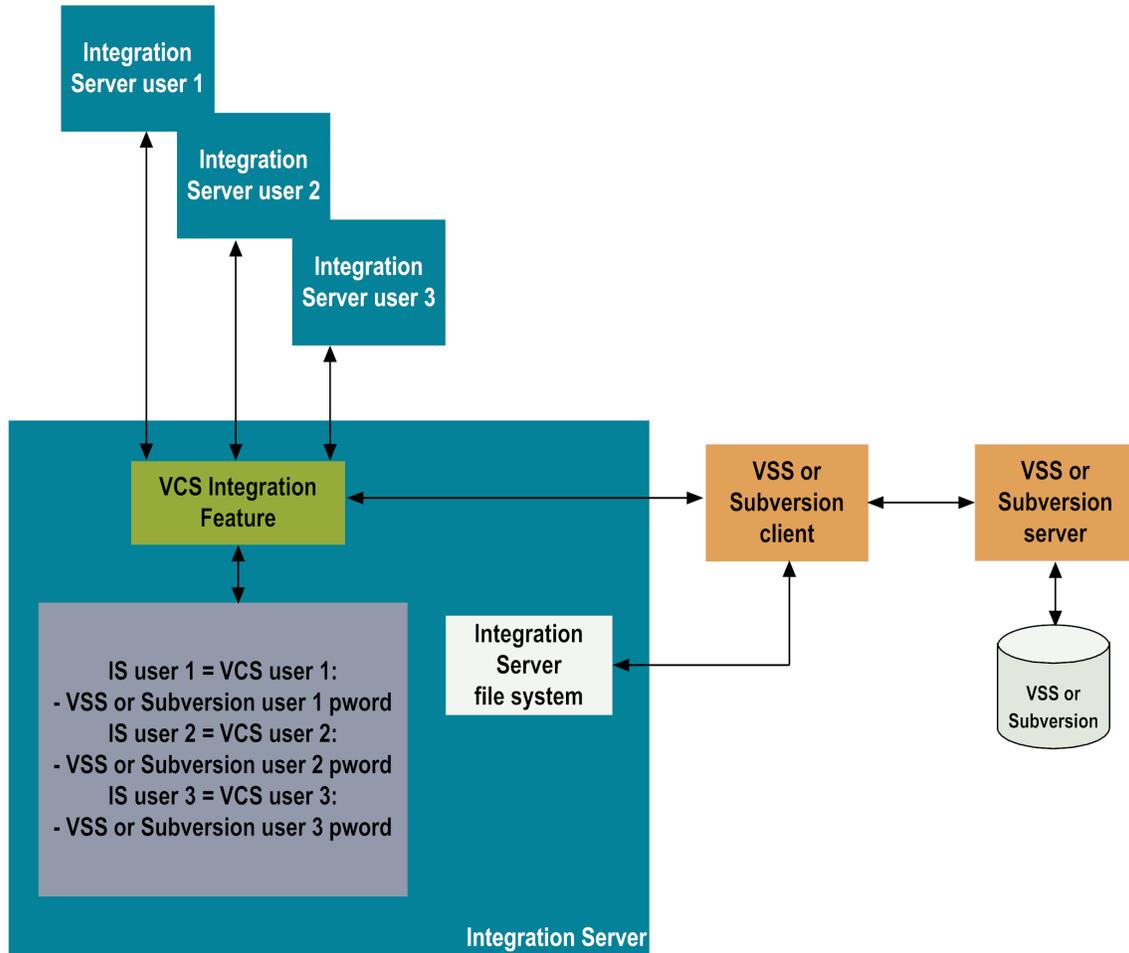
Important: This section applies to Visual SourceSafe and Subversion only.

Although each Integration Server can connect to only one VCS server, in many cases two or more Designer users will be working on the same Integration Server, perhaps working on the same package or packages at different times. In this shared development environment, it is imperative to know which Designer user committed a particular change to the VCS repository.

When a VCS transaction is entered into the version history on the VCS server, both the Designer user and the VCS user are recorded for the transaction. For example, when you apply a **Check In** command in Designer to a package, folder, or element, the underlying files are checked in and the version history records both the Designer user name and the VCS user name.

For Visual SourceSafe and Subversion, as described in [“Configuring Access to the VCS Repository” on page 39](#), the VCS Integration feature lets you to map your Integration Server user name to a Visual SourceSafe or Subversion user, respectively. If an Integration Server user is not mapped to a Visual SourceSafe or Subversion user, all Visual SourceSafe or Subversion commands applied in Designer will fail (except as noted in [“Working with a Windows System” on page 40](#)). It is not possible to revert to the basic locking behavior at the individual user account level in Designer.

The following diagram illustrates the behavior of multiple users mapped to Visual SourceSafe or Subversion users with the VCS Integration feature. In this diagram, each of three different users is logged on to Integration Server. The VCS Integration feature recognizes each Integration Server user as a unique user. Through use of the VCS Integration feature's mapping feature, each of the Integration Server users has identified the Visual SourceSafe or Subversion user that is to be specified to the Visual SourceSafe or Subversion server when that user checks files in to or out of the Visual SourceSafe or Subversion repository.



5 Security Considerations

■ General Security Considerations	56
■ About VCS Users	56
■ Checking of External Input	57
■ Validity of VCS History Information	58

General Security Considerations

The VCS Integration feature works within the native Integration Server or Designer security model. For example, you must be logged on as a valid Integration Server or Designer user to gain access to the VCS Integration commands, and Access Control Lists (ACLs) must be configured to allow you access to the packages, folders, and elements you want to work with in Designer.

Furthermore, a user cannot apply VCS commands to any Integration Server contents other than the packages, folders, and elements displayed within the Package Navigator view of Designer.

VCS commands are applied at the package, folder, and element level only; the user does not directly apply VCS commands to individual files, nor does the user have visibility of individual files in the VCS repository through Designer. Therefore it is not possible for a Designer user to manipulate files in the VCS repository using the VCS Integration feature unless that user has ACL access in Designer to the package, folder, or element containing those files. In this case, the files are manipulated indirectly, as children of the selected package, folder, or element.

Users must be able to log on to the Visual SourceSafe, ClearCase, or Subversion repository with authentication credentials that are different from the Integration Server authentication credentials.

About VCS Users

User Mappings for Visual SourceSafe and Subversion

Integration Server users gain access to the Visual SourceSafe or Subversion repository by being mapped to a Visual SourceSafe or Subversion user name and password. Each Integration Server user can map him or herself to a Visual SourceSafe or Subversion user. Integration Server users with administrator privileges can create mappings for other Integration Server users. For more information, see [“Configuring Access to the VCS Repository” on page 39](#).

The VCS Integration feature uses Visual SourceSafe or Subversion user credentials to interact with the Visual SourceSafe or Subversion repository; this means the Visual SourceSafe or Subversion user credentials must be stored on Integration Server. The VCS Integration feature encrypts the password portion of the credentials and stores the passwords in encrypted format.

The VCS Integration feature maintains the user mappings in the `users.cnf` file in the `Integration Server_directory\instances\instance_name\packages\WmVCS\config` directory. However, anyone with write access to the `users.cnf` file can open the file and modify the mappings. Because the VCS Integration feature encrypts the passwords for the Visual SourceSafe or Subversion users, a malicious user cannot gain access

to the Visual SourceSafe or Subversion repository by defining a new mapping in the `users.cnf` file, but operation of the VCS Integration feature can be disrupted if entries in the `users.cnf` file are deleted or corrupted. It is recommended that you set permissions on the `users.cnf` file so that only trusted users have write access to the file.

VCS Commands

The VCS Integration feature requires that the VCS client be installed on the Integration Server host machine and connected to the VCS server. When a VCS command is applied to a package, folder, or element in the Package Navigator view in Designer, the VCS Integration feature passes a command to the VCS client for each underlying file contained in the selected item. This command contains the VCS user credentials. After the command is received by the VCS client, the VCS client transmits it to the VCS server using the security methods inherent to the VCS client and VCS server. An Integration Server user who has permission to run the VCS client can also connect to the VCS repository directly through the VCS client software.

Permission to run the VCS client software is required for each Integration Server user who will work with the VCS Integration feature. Consider prohibiting Integration Server users who do not need to access the VCS repository from running the VCS client. You should also limit VCS users to the minimum file permissions and system rights required to carry out the work assigned to those users.

Logging

The VCS Integration feature provides logging in compliance with Integration Server standards, writing to the appropriate log file in the *Integration Server_directory*\instances*instance_name*\log directory. No user credentials are logged with the default log settings. However, when more detailed log settings are prescribed by Software AG Global Support, user credentials might be contained in the log entries.

In these situations, the user name is displayed, but the password is presented as asterisks. For example:

```
executing ss -YuserName,***** checkout  
$/WmPkg/code/source/proj/service/file.java
```

Checking of External Input

Because of the requirement to interact with an external VCS server, the VCS Integration feature must necessarily accept data transmitted from the VCS server. For example, within the VCS server, it is possible for a malicious user to create a version label containing a destructive command. If the same malicious user logged on to Designer and used the **Get Earlier Version** command to retrieve a package or element tagged with this label, the command could damage the Integration Server file system.

The VCS Integration feature minimizes this threat by checking data arriving from the VCS server for the prohibited characters ";" and "&". If either of these characters is found, the operation is halted and an error message appears within Designer.

Validity of VCS History Information

When a VCS check in transaction originating through the VCS Integration feature is entered into the version history on the VCS server, the user can enter a comment. By default, if no comment is entered, the Designer user name, Integration Server host name and IP address, and current time are entered in place of the comment. This information is determined by two entries in the `vcs.cnf` file in the `Integration Server_directory\instances\instance_name\packages\WmVCS\config` directory.

There is one entry for the **Check In** command, and one entry for the **Delete** command. Each entry passes the current Designer user name and other information to the VCS client, which in turn sends it to the VCS server.

Anyone with write access to the `vcs.cnf` file can open and delete this information, or modify these entries to pass information other than the current Designer user, host name, IP address, and time for the default entry for check in and delete commands. It is recommended that you ensure that permissions are set on this configuration file so that only trusted users have write access to the file.

A Troubleshooting

■ Cannot Connect to the VCS Server	60
■ VCS Menu Commands Are Not Present	61
■ VCS Menu Commands Are Not Available	61
■ An "Access Denied" Exception Occurs When Saving an Element	62
■ Cannot Create a C/C++ Service	62
■ Cannot Get Latest Version of a C Service	62
■ Designer Loses Connection with Integration Server During Large Transactions	63
■ Designer Appears to Stop Responding After Applying a VCS Command	63
■ A Check-In Comment Is Not Recorded in the Revision History (Visual SourceSafe)	64
■ VCS Transactions Result in Unexpected Behavior (Visual SourceSafe)	65

Cannot Connect to the VCS Server

The VCS Integration feature commands are present and available in Designer, but selecting an element and clicking a VCS command results in an error message.

Possible Cause	Solution
The VCS client is not currently installed, or is unavailable, on the Integration Server computer.	Attempt to start the VCS client application manually. If the VCS client is not present, it must be installed and configured prior to using the VCS Integration feature. The VCS client application must be available to enable interaction with the VCS server application.
The VCS client is not executable by the Designer user account.	The VCS client application must be executable by the Designer user account to enable interaction with the VCS server application. Ensure that each Designer user account that will be working with the VCS Integration feature has permission to run the VCS client on Integration Server.
The VCS server application is not running.	Start the VCS server program. Test the ability to connect to the VCS server by starting your VCS client.
Network connectivity has failed between the Integration Server computer and the VCS server.	Re-establish the network connection. Test the ability to connect to the VCS server by starting your VCS client. Be sure to check with your network or system administrator to determine if the network configuration has been changed without your knowledge.
The VCS client is not properly configured (the host name, port, client name, SSDIR variable, or project name is missing or incorrect, for example).	Consult your VCS client documentation and configure the VCS client as needed to connect to the VCS server. Be sure to check with your network or system administrator to determine if a host name or port assignment has been changed without your knowledge.
The VCS-specific package is not properly configured.	Review the procedures described in “Configuring the WmVCS Package to Work with Visual SourceSafe” on page 35 and ensure that the <code>VCSname.cnf</code> file in the following directory has the correct entries:

Possible Cause	Solution
	<p><i>Integration Server_directory</i> \instances <i>\instance_name</i> \packages \WmVCSname</p> <p>Entries in the vcs.cnf file are case-sensitive. Modify the settings as required, and save the file. Log on to the Integration Server Administrator and disable and then enable the package to apply the changes. Be sure to check with your network or system administrator to determine if a host name or port assignment has been changed without your knowledge.</p>

VCS Menu Commands Are Not Present

The VCS menu commands do not appear on the **File** and pop-up menus. In their place are the standard **Lock** and **Unlock** menu commands.

Possible Cause

The packages supporting the VCS Integration feature are not installed or are disabled on the Integration Server.

Solution

In Integration Server Administrator, go to the **Packages > Management** page. Make sure the WmVCS package, which provides the VCS Integration feature, is enabled. Make sure the package for your VCS server (WmSourceSafe, WmClearCase, or WmSubversion) is installed and enabled; if it is not, install and enable it. For instructions, see [“Enabling and Disabling the VCS Integration Packages”](#) on page 42.

VCS Menu Commands Are Not Available

Some VCS menu commands are present but unavailable (grayed out) when a package, folder, or element is selected.

Possible Cause

The unavailable menu commands do not apply to the selected package, folder, or element. For example:

- The **Revert** command is not available unless an element is checked out
- The **Get Earlier Version** and **Get Latest Version** commands are not available for checked out elements
- The **View History** command is not available for packages

Although folders do not display the checked out icon, they do maintain a checked in or checked out status. It is possible that checked out elements within the folder will cause menu commands to be unavailable for a folder. It is also possible that all of the folders

and elements in the folder's hierarchy have been checked in individually, but the folder is still in a checked out state.

Solution

Examine the contents of the folder for checked out elements. You can apply the **Revert** or **Check In** commands to the elements or to the folder to either discard or commit any changes made to elements in the folder's hierarchy.

An "Access Denied" Exception Occurs When Saving an Element

When you check out a package, folder, or element and make changes to it, and then save the changes in Designer, an exception error occurs stating that access is denied. You are unable to save the changes or check in the element.

Possible Cause

The **Automatically unlock upon save** option is enabled in Designer. This option conflicts with VCS Integration feature operation and must be disabled at all times when the VCS Integration feature is enabled.

Solution

Disable the option as described in ["Configuring Integration Server and Designer"](#) on page 34.

Cannot Create a C/C++ Service

When you try to create a C/C++ service, you receive the error, "Could not properly generate C service."

Possible Cause

All C/C++ services must be uniquely named, regardless of their location in the Navigation panel.

Solution

Create the service with a unique name.

Cannot Get Latest Version of a C Service

When you try to get latest on a C service, you receive the error, "The following elements were not loaded with the current version."

Possible Cause

The C Service has a corresponding uncommitted Specification file.

Solution

Commit the Specification file and retry.

Designer Loses Connection with Integration Server During Large Transactions

After selecting a large number of elements and applying an action such as Copy, Designer loses its connection with the Integration Server

Possible Cause

This issue is resource dependent; that is, it might be caused by limitations in CPU speed, available memory, hard drive space, network performance, and so on.

Solution

Apply the action to smaller sets of elements in succession.

Designer Appears to Stop Responding After Applying a VCS Command

Designer appears to stop responding after you apply a VCS command to a package, folder, or element. Designer displays an error message that the time-out period was exceeded (in cases where the time-out feature is disabled, no error message is displayed).

Possible Cause

The VCS user lacks the required permissions to work with the specified files. This situation is limited to the Integration Server user who has submitted VCS credentials with incorrect permissions. Other Designer users working on the same Integration Server will not be affected.

Solution

If you have not received a time-out error, return normal operation of Designer by stopping the VCS client. For Visual SourceSafe, press CTRL+SHIFT+ESC to open the Windows Task Manager. On the **Processes** tab, locate and select the SS.EXE process, and then click **End Process**.

Modify the VCS user using the VCS server's user management feature. If no time-out exceeded error message appears, specify a time-out value as described in ["Configuring the WmVCS Package to Work with Visual SourceSafe" on page 35](#).

Note that all VCS users must have read and write permission for the files they are working with.

Possible Cause

The VCS user credentials specified for use by the VCS Integration feature are invalid.

This situation is limited to the Designer user who has submitted the incorrect VCS credentials. Other Designer users working on the same Integration Server will not be affected.

Each Integration Server user must be mapped to a valid VCS user. When you apply a VCS command to a package, folder, or element in Designer, the command is passed to the VCS client on the Integration Server system. The VCS client passes the command and the credentials to the VCS server. If the user name or password is invalid, the VCS server notifies the VCS client.

When operating in command-line mode, the Visual SourceSafe client provides no means of notifying the user that it requires corrected values for the user name or password. As a result, the Visual SourceSafe client waits for input that has not been requested. Designer, in turn, is waiting for the Visual SourceSafe client to complete the transaction. If you have not specified a time-out period for the VCS Integration feature, the VCS client will wait indefinitely.

Solution

If you have not received a time-out error, you must stop the Visual SourceSafe client. To do so, press CTRL+SHIFT+ESC to open the Windows Task Manager. On the **Processes** tab, locate and select the SS.EXE process, and then click **End Process**. This should return normal operation of Designer.

You must now enter the correct VCS user name and password to be used by your Integration Server user as described in [“Mapping One Integration Server User to One VCS User”](#) on page 40.

If no time-out exceeded error message appears, specify a time-out value as described in [“Configuring the WmVCS Package to Work with Visual SourceSafe”](#) on page 35.

A Check-In Comment Is Not Recorded in the Revision History (Visual SourceSafe)

When you check in a package, folder, or element, a Check In Comment dialog box appears, enabling you to type a comment on the nature of the check in. When you view the revision history for the element, the comment is not present.

Possible Cause

In Visual SourceSafe, the default check in behavior is to cancel the check in for unchanged files. Therefore, Visual SourceSafe disregards the comment you entered when you check in an unchanged file. The comment is applied when you check in a modified file.

Solution

Configure Visual SourceSafe to check in unchanged files. In Visual SourceSafe 6.0, this is done from the **Tools > Options** menu; on the SourceSafe Options dialog box, click the **General** tab. Set the value for **Check in unchanged files** to Check In.

VCS Transactions Result in Unexpected Behavior (Visual SourceSafe)

After checking in or checking out elements from the VCS repository, unexpected behavior results. For example, user names in the revision history are incorrect, VCS repository options are not applied, or transactions fail.

Possible Cause

This issue is mostly likely caused by a conflict in user names being provided to the VCS client running on the Integration Server. If the VCS client is not running at the time of a VCS transaction in Designer, the VCS Integration feature starts the VCS client and passes it the appropriate command string, which includes the VCS user account name associated with the current Designer user account. When the transaction is complete, the VCS client terminates. This mode of operation causes no problems. However, it is possible that the VCS client may already be running and connected to the VCS server with a VCS user account that is different from the VCS user account that is associated with the current Designer user. In this case, when a VCS transaction occurs, the VCS Integration feature will pass the VCS client a VCS user name that is different than the VCS client's connected user name. This causes unexpected results in the VCS transaction.

Example

Patrick starts the VCS client on Integration Server and connects to the VCS server as user Patrick_VCS. He completes his VCS tasks but leaves the VCS client running. Patrick then logs on to Designer as user Administrator. This account is associated with the VCS user account Admin_VCS. Patrick attempts to check out some elements from the VCS. The VCS Integration feature submits the transaction to the VCS client with the VCS user name of Admin_VCS, but the VCS client is connected to the server as Patrick_VCS. Unexpected results occur.

Solution

Either shut down the VCS client on the Integration Server, or restart the VCS client and connect to the VCS server using the same VCS user name as the one associated with the current Designer user.