

Oracle® Fusion Middleware
Site Studio Installation Guide
10g Release 4 (10.1.4.5.0)

May 2010

Copyright © 1996, 2010, Oracle and/or its affiliates. All rights reserved.

Primary Author: Sean Cearley

Contributor: Brian Cheyne, Ron van de Crommert

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	vii
Audience	vii
Documentation Accessibility	vii
Related Documents	viii
Conventions	viii
1 Introduction	
1.1 About Site Studio	1-1
1.2 System Requirements	1-1
1.2.1 Requirements for Content Server	1-2
1.2.2 Requirements for the Designer Application	1-3
1.2.3 Requirements for the Manager Application	1-3
1.2.4 Requirements for the Contributor Application	1-3
1.3 Installation Process	1-3
1.4 Documentation	1-4
2 Installing the Site Studio Component	
2.1 Pre-Installation Tasks and Considerations	2-1
2.2 Uninstalling the Previous Component	2-1
2.3 Installing and Enabling the New Component	2-2
2.4 Post-Installation Tasks and Considerations	2-3
2.4.1 Setting Default Project Document Information	2-4
2.4.2 Configuring Apache Web Server	2-5
2.4.3 Configuring Sun ONE Web Server	2-6
2.4.4 Updating the SUrlMapPlugin File	2-6
2.4.5 Configuring Zone Fields	2-7
2.4.6 Enabling JavaServer Pages on the Content Server	2-7
2.4.7 Configuring Active Server Pages on the Content Server	2-8
2.4.8 Configuring Active Server Pages in a Clustered Content Server Environment	2-9
2.5 Rebuilding the Content Server Index	2-10
2.6 Restarting the Web Server	2-11
3 Installing Designer	
3.1 About Designer	3-1
3.2 System Requirements	3-1

3.3	Installing Designer.....	3-2
3.4	Launching Designer.....	3-2

4 Setting Up Contributor

4.1	About Contributor	4-1
4.2	System Requirements	4-1
4.3	Launching Contributor	4-1

5 Installation Considerations

5.1	Site Studio Metadata Fields	5-1
5.1.1	Web Sites	5-1
5.1.2	Web Site Section	5-2
5.1.3	Web Site Object Type	5-2
5.1.4	Exclude From Lists	5-3
5.1.5	Region Definition.....	5-4
5.2	Changing the Default Contributor Keystroke Combination	5-4
5.3	Blocking Access to the Consumption Server	5-5
5.4	Changing the Default Contributor Editor	5-5
5.5	Updating Custom Elements	5-5
5.6	Using the Certificate Validation Applet	5-6
5.7	Configuring a Single Sign-On (SSO) Environment.....	5-7
5.8	Enabling Debugging in Contributor	5-8
5.9	Using Pre-10gR4 Site Studio Projects	5-8

A Uninstalling Site Studio Software

A.1	Uninstalling Designer.....	A-1
A.2	Uninstalling the Site Studio Component.....	A-1

B Upgrading Pre-7.5 Web Sites

B.1	Introduction	B-1
B.2	What the Automated Upgrade Does.....	B-1
B.3	Upgrading Your Content Servers.....	B-2
B.3.1	Upgrading Sites on a Single Content Server Instance.....	B-3
B.3.2	Upgrading Sites on Multiple Content Server Instances.....	B-3
B.3.3	Performing a Full Upgrade	B-4
B.3.4	Performing a Minimal Upgrade	B-5
B.4	Performing Additional Steps Manually	B-6
B.4.1	Updating the Site Navigation	B-6
B.4.2	Rebuilding the Content Server Index	B-6
B.4.3	Updating Your Custom Fragments.....	B-7
B.4.3.1	Modifying Links That Rely on the <base> Tag.....	B-7
B.4.3.2	Modifying Obsolete SS_GET_PAGE / JavaScript Links	B-7
B.4.3.3	Updating GET_SEARCH_RESULTS	B-7
B.4.4	Updating Your Custom Elements	B-10
B.4.5	Assigning a Web Site Section to Your Folders	B-10

B.4.6	Updating JSP Code	B-11
-------	-------------------------	------

Index

Preface

The Site Studio Installation Guide contains information to assist administrators responsible for setting up a Site Studio environment. This guide provides introductory information, configuration notes, and installation instructions.

Audience

This document is intended for system administrators. It includes information specific to installing and setting up the Site Studio software.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/support/contact.html> or visit <http://www.oracle.com/accessibility/support.html> if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Site Studio documentation set (see also "[Documentation](#)" on page 1-4):

- *Oracle Site Studio Contributor Guide*
- *Oracle Site Studio Designer Guide*
- *Oracle Site Studio Administrator and Manager Guide*
- *Oracle Site Studio Release Notes*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Introduction

This section covers the following topics:

- ["About Site Studio"](#) on page 1-1
- ["System Requirements"](#) on page 1-1
- ["Installation Process"](#) on page 1-3
- ["Documentation"](#) on page 1-4

Important: There are major architectural changes in Site Studio 10gR4 compared to earlier releases, including 10gR3. The concepts and site design process are entirely different with this new release. You may want to read the first few chapters of the Site Studio Designer Guide for an overview of the new architecture and concepts of the Site Studio 10gR4 release.

1.1 About Site Studio

Site Studio is a powerful, flexible web development application suite that offers a comprehensive approach to designing, building, and maintaining enterprise-scale Web sites. It goes beyond conventional Web site development solutions by offering Web site creation and content management all in one. Everything associated with the Web site, including all site assets (such as templates, graphics, CSS files, and the like) and all site content, is stored and managed in the content server. Site Studio dramatically improves productivity and allows organizations to maintain accurate, timely, and current web content with consistent branding and presentation across all corporate sites. With Site Studio, you can centralize control of site architecture and presentation, while distributing content development and ongoing maintenance to business units or other teams.

1.2 System Requirements

Site Studio consists of one Content Server component and three applications, each with its own set of system requirements:

- ["Requirements for Content Server"](#) on page 1-2
- ["Requirements for the Designer Application"](#) on page 1-3
- ["Requirements for the Manager Application"](#) on page 1-3
- ["Requirements for the Contributor Application"](#) on page 1-3

1.2.1 Requirements for Content Server

Content Server

Site Studio 10gR4 can be used with the following Content Server releases:

- **Content Server 10gR3**, with the following software components:
 - SiteStudio component
 - DBSearchContainsOpSupport component (see note below about indexing)
 - CS10gR3CoreUpdate patch (CS 10gR3 version 10.1.3.3.2 or earlier only)
 - CS10gR3NativeUpdate patch (CS 10gR3 version 10.1.3.3.2 or earlier only)
- **Content Server 7.5.2**, with the following software components:
 - SiteStudio component
 - DBSearchContainsOpSupport component (see note below about indexing)
 - CS752Update patch

The SiteStudio component is included in the Site Studio 10gR4 distribution package. If you plan to view your site using a Content Server proxy, make sure that you install the Site Studio component on both the Content Server proxy and its associated master. Otherwise, the site cannot be viewed successfully.

The DBSearchContainsOpSupport component is typically installed along with the Content Server software. If this is not the case, you can find the component zip file in the `\packages\allplatform` directory of the Content Server distribution package.

The latest versions of the required update patches are available from Oracle Metalink (<http://metalink.oracle.com>).

Content Server Add-Ons

Site Studio works with several other Content Server add-ons:

- **Dynamic Converter**, which converts native documents used in your Web site to HTML, so they can be viewed in a web browser.
- **Check Out and Open**, which allows contributors to check native documents out of the content server from Site Studio and open them in their associated third-party application for editing.
- **Content Tracker / Content Tracker Reports**, which allows contributors to view site reports that show how many times files on the site were viewed and who viewed those files.

If you are using any of these add-ons, their version must be compatible with the version of your content server. (For more information on these components, see the Content Server documentation.)

Indexing

The content server may be set up for full-text indexing or metadata-only indexing. However, it is crucial that the DBSearchContainsOpSupport component is installed on the content server. This component ensures that some important Site Studio metadata fields are full-text indexed properly, which is required for the sites to work correctly.

If you are using database search and indexing (full-text or metadata-only), you do not need to rebuild the search index when installing or upgrading the Site Studio component on the content server. If you are using a different search engine (typically,

Verity), you must rebuild the search index when installing or upgrading the Site Studio component. See ["Rebuilding the Content Server Index"](#) on page 2-10 for more information.

Folders

The Content Server's folders functionality is not required by Site Studio. If, however, you are upgrading from a Site Studio release before 7.5, the folders functionality must be enabled to upgrade folder-based sites to project-based sites. After upgrading the sites, you can then disable folders (assuming you are not using other components that rely on the folders functionality). See [Appendix B, "Upgrading Pre-7.5 Web Sites"](#) for more information on the upgrade process.

If you want to continue using folders with your site, you must assign different metadata to them so that the content added there is recognized as part of your site.

1.2.2 Requirements for the Designer Application

The system requirements for the Site Studio Designer application are as follows:

- Windows XP (SP2 or higher), Windows Vista (SP1 or higher), Windows 2003 Server (SP2 or higher), Windows 2003 Server R2, Windows Server 2008 (SP1 or higher), Windows 7 operating systems.
- Access to an Oracle Content Server instance running the Site Studio component.
- Microsoft Internet Explorer 5.5 or higher. (You can use Microsoft Internet Explorer 5.5 or higher, or Mozilla Firefox 1.0.7 or higher to view the resulting web pages.)

1.2.3 Requirements for the Manager Application

The system requirements for the Site Studio Manager application are as follows:

- Access to the Oracle Content Server running the Site Studio component.
- Microsoft Internet Explorer 5.5 or higher, or Mozilla Firefox 1.0.7 or higher.

1.2.4 Requirements for the Contributor Application

The system requirements for the Site Studio Contributor application are as follows:

- An operating system that can run Internet Explorer 6.0 or higher, or Firefox 2.0 or higher (for example, Microsoft Windows 2000, Windows XP, Windows Vista, Windows 7, Linux, or Mac OS X).
- Access to an Oracle Content Server instance running the Site Studio component.
- Microsoft Internet Explorer 6.0 or higher, or Firefox 2.0 or higher. To view the resulting web pages, you need Microsoft Internet Explorer 5.5 or higher, or Mozilla Firefox 1.0.7 or higher.

1.3 Installation Process

Installing and setting up Site Studio consists of the following steps:

1. Make sure that Oracle Content Server is running correctly and has been updated with all required patches, where necessary. See ["Requirements for Content Server"](#) on page 1-2.

2. Make sure that the DBSearchContainsOpSupport component is installed and enabled in the content server. See the Content Server installation documentation for instructions.
3. Upload and enable the Site Studio component on the content server. For more information, see [Chapter 2, "Installing the Site Studio Component."](#)
4. Install the Designer application on the system that will be used to create and design Web sites. For more information, see [Chapter 3, "Installing Designer."](#)

You can install everything on the same system, but you will likely use a dedicated server for the content server (and therefore the component) and multiple systems for design contribution.

See also [Chapter 5, "Installation Considerations"](#) for several post-installation tasks and considerations.

1.4 Documentation

The following documentation is available for Oracle Site Studio 10gR4:

Document	Availability
Release Notes	Available as a PDF file in the Site Studio software distribution package.
Installation Guide	Available as a PDF file in the <i>Documentation</i> folder of the Designer installation directory and the Site Studio software distribution package.
Designer Guide	Available from the Help menu and Help dialog buttons in the Site Studio Designer application. The PDF version (useful for printing the documentation) is available in the <i>Documentation</i> folder of the Designer installation directory and the Site Studio software distribution package.
Administrator and Manager Guide	Available from the Help links in the Site Studio Manager application and the Site Studio administration pages (in the Content Server user interface). The PDF version (useful for printing the documentation) is available in the <i>Documentation</i> folder of the Designer installation directory and the Site Studio software distribution package.
Contributor Guide	Available from the Help links in the Site Studio Contributor application. The PDF version (useful for printing the documentation) is available in the <i>Documentation</i> folder of the Designer installation directory and the Site Studio software distribution package.

Installing the Site Studio Component

The Site Studio component is installed on the content server. This process consists of the following steps:

- ["Pre-Installation Tasks and Considerations"](#) on page 2-1
- ["Uninstalling the Previous Component"](#) on page 2-1
- ["Installing and Enabling the New Component"](#) on page 2-2
- ["Post-Installation Tasks and Considerations"](#) on page 2-3
- ["Rebuilding the Content Server Index"](#) on page 2-10
- ["Restarting the Web Server"](#) on page 2-11

2.1 Pre-Installation Tasks and Considerations

Before installing the Site Studio component, you should verify the following:

- The Site Studio distribution package has been uncompressed to a temporary location on your hard drive.
- The content server meets all system requirements for Site Studio 10gR4. See ["Requirements for Content Server"](#) on page 1-2 for details.
- The required software patches are installed on the content server. See ["Requirements for Content Server"](#) on page 1-2 for details.
- The DBSearchContainsOpSupport component is installed and enabled on the content server. See the Content Server installation documentation for instructions.
- If you are upgrading from an earlier Site Studio version, the previous component has been uninstalled. For more information, see ["Uninstalling the Previous Component"](#) below.

Important: If you are upgrading your Site Studio installation and want to keep Ephox as your editor, you should not uninstall the Site Studio component.

2.2 Uninstalling the Previous Component

If you are upgrading from a previous version of Site Studio, uninstall the previous component before installing the new component:

Important: Site Studio 10gR4 (10.1.4.5.0, build 611 and higher) does not include Ephox as an editor. If you want to continue to use Ephox as your editor, then do not uninstall the previous version. Install the new component on top of the old component.

1. Open a new browser window and log in to Content Server as a system administrator (with the "sysmanager" role).
2. Go to the Administration Applets page and click the **Admin Server** link.
3. On the Content Admin Server page, click the button of the content server instance that you want to uninstall the Site Studio component from.
The status page for the content server instance is displayed.
4. In the option list for the server instance, click the **Component Manager** link.
The Component Manager page is displayed.
5. Under Enabled Components, select **SiteStudio**.
6. Click **Disable**.
The SiteStudio component is moved to the list of disabled components.
7. Restart the content server.
8. Under Uninstall Component, select **SiteStudio**.
9. Click **Uninstall**.
10. Restart the content server.

2.3 Installing and Enabling the New Component

To install and enable the Site Studio component, perform these tasks:

1. Open a new browser window and log in to Content Server as a system administrator (with the "sysmanager" role).
2. Go to the Administration Applets page and click the **Admin Server** link.
3. On the Content Admin Server page, click the button of the content server instance that you want to install the Site Studio component on.
The status page for the selected content server instance is displayed.
4. In the option list for the server instance, click the **Component Manager** link.
The Component Manager page is displayed.
5. Click the **Browse** button next to the **Install New Component** field.
6. Navigate to the Site Studio component zip file (in the *Component* directory of the software distribution package), select it, and close the file selection dialog.
7. Click **Install**.
An overview page is displayed providing a list of the items that will be installed.
8. Click **Continue**.
The Install Settings page is displayed.
9. Select a content type for the fragment libraries, custom element forms, validation scripts, sample Web site objects, and custom configuration scripts.

10. Enter an initial value to be used to name the Web site sections. This value is used to create unique IDs for the nodes (sections) of your Web site. If you are installing the component on a single content server, you can accept the default. If you are installing the component on more than one content server, you may want to use different initial values to distinguish between the Web sites stored in the different servers.
11. Click **Continue**.
All required files are now uploaded and installed.
The upload may take a long time, especially when installing onto a network drive, and there is no indication of progress.
After all files have been copied, a message is displayed stating that the component was uploaded and installed successfully.
12. Click the link to enable the component and restart the server.
The content server status page is displayed.
13. Click the restart icon ([Figure 2–1](#)) to restart the content server instance.

Figure 2–1 Restart Icon



All of the fragments and sample files you need to run Site Studio are automatically checked into the content server during installation. New metadata fields specific to Site Studio are also added to the content server.

If you plan to view your site using a content server proxy, you must install the component on the master instance of the content server as well.

2.4 Post-Installation Tasks and Considerations

After installing the new Site Studio component, you should note the following post-installation tasks and considerations:

- Make sure that you set the default project document information. This is necessary for the project-based hierarchy to work. For more information, see "[Setting Default Project Document Information](#)" on page 2-4.
- If you are using Apache as the web server, the server must be configured so that Site Studio can handle path-based URLs. For more information, see "[Configuring Apache Web Server](#)" on page 2-5.
- If you are using Sun ONE as the web server, the server must be configured to include the web ID in its NameTrans configuration entry. For more information, see "[Configuring Sun ONE Web Server](#)" on page 2-6.
- The SSUrlMapPlugin file may need to be updated. The Site Studio component will replace the SSUrlMapPlugin file during installation, but you may need to update it manually in certain situations. For more information, see "[Updating the SSUrlMapPlugin File](#)" on page 2-6.
- You must designate some Site Studio metadata fields as zone fields to ensure they can be full-text searched. For more information, see "[Configuring Zone Fields](#)" on page 2-7.

- If you will use JavaServer Pages (JSP) with Site Studio, JSP must be enabled on the content server. For more information, see ["Enabling JavaServer Pages on the Content Server"](#) on page 2-7.
- If you will use Active Server Pages (ASP) with Site Studio, ASP support must be enabled on IIS and the server must be configured to use relative paths. For more information, see ["Configuring Active Server Pages on the Content Server"](#) on page 2-8 (for non-clustered content server environments) or ["Configuring Active Server Pages in a Clustered Content Server Environment"](#) on page 2-9 (for clustered environments).

Important: JSP and ASP are supported only in legacy Site Studio projects; that is, projects that use the pre-10gR4 architecture. These are typically projects that were created in a Site Studio release before 10gR4 and that are opened in Designer 10gR4.

- If you are upgrading from a Site Studio release before 7.5, you must upgrade all of the Web sites that you created with the previous version. For more information, see [Appendix B, "Upgrading Pre-7.5 Web Sites."](#)
- If you are upgrading from version 7.5 or 10gR3, you must update the site navigation, which can be done in the Site Studio Designer application and on the Manage Web Sites page in Content Server.

Important: Make sure that you restart the web server after setting up Site Studio.

- Site Studio no longer contains Ephox. However, it is possible to upgrade to the 10.1.4.5.0 component and retain Ephox as the editor. To do this, do not uninstall the previous component version. Install the new Site Studio component on top of the old component to retain Ephox as the editor.

2.4.1 Setting Default Project Document Information

When you create a new Web site in Site Studio, a new project file is created and checked into the content server for you. Thus, before you can create Web sites, you must specify the metadata that will be assigned to the new project files. You do this on the Set Project Default Document Information page in the content server.

To set the default metadata used for Site Studio project files, perform these tasks:

1. Log on to the content server as an administrator.
2. Go to the **Administration** page, and click **Site Studio Administration**.
The Site Studio Administration page is displayed.
3. Click **Set Default Project Document Information**.
The Set Default Project Document Information page is displayed, where you assign the default metadata for new projects generated by Site Studio.
4. Set the metadata as required, and click **Update** when you have finished.
This returns you to the Site Studio Administration page.

2.4.2 Configuring Apache Web Server

If you are using an Apache web server as your web server, you should update the configuration file to take advantage of the path-based URLs in Site Studio. If you are not using Apache as your web server, you can skip this section.

To edit the Apache web server configuration file, perform these tasks:

1. Open the Apache *httpd.conf* configuration file, which is located in the *conf* directory of Apache's installation location.
2. Locate the entries that look something like this:

```
LoadModule IdcApacheAuth [CS-Dir]/shared/os/[OS_Name]/lib/IdcApacheAuth2.dll
IdcUserDB myserver [CS-Dir]/data/users/userdb.txt
Alias /myserver "[CS-Dir]/weblayout"
<Location "/myserver">
    DirectoryIndex portal.htm
    IdcSecurity myserver
</Location>
```

Note: On UNIX, the LoadModule line refers to IdcApacheAuth2.so.

3. Add the following lines:

```
<Location "/">
    IdcSecurity myserver
</Location>
```

4. Then locate the "UseCanonicalName" configuration variable and make sure it is set to **Off**.
5. Save the configuration file and restart Apache HTTP Server.

Notes

- In all of these steps, the name of the Content Server instance is 'myserver'. If you named your Content Server something else, then there will be slight differences. For example, if your Content Server is named "cherokee," then step 3 would be as follows:

```
<Location "/">
    IdcSecurity cherokee
</Location>
```

All code examples use "myserver" as the Content Server name.

- If you want to use Site Studio domain-based sites, then you must use the <Location "/"> entry (in step 3).
- If you prefer not use the <Location "/"> entry (in step 3), then you must use a separate entry for each of the Web sites that the server should recognize. For example, if you have two sites with the addresses "http://<domain>/site1/index.htm" and "http://<domain>/site2/index.htm," then you could set up two location entries as follows:

```
<Location "/site1">
    IdcSecurity myserver
</Location>
<Location "/site2">
    IdcSecurity myserver
</Location>
```

For information on changing Web site URLs in Site Studio, see in the *Site Studio Designer Guide*.

2.4.3 Configuring Sun ONE Web Server

If you are using a Sun ONE web server as your web server, you should update the configuration file to take advantage of the path-based URLs in Site Studio. If you are not using Sun ONE as your web server, you can skip this section.

To edit the Sun ONE web server configuration file, perform these tasks:

1. Go to the Sun web server software directory, and open the *https-[host_name]/config* subdirectory (where *[host_name]* is the name of the system on which the software is installed), for example:

```
/https-server7/config
```

2. Open the file *obj.conf* in a text editor.
3. Inside the `<Object name="default">` section, add the web id to the NameTrans entry:

```
NameTrans fn="pfx2dir" from="/[Site_ID]" dir="[Weblayout_Dir]"
```

where *[Site_ID]* is the web id of your Web site (for example `/Xalco,` and *[Weblayout_Dir]* is the full path to the web-viewable file repository (for example, `/ul/cserver/idcm1/weblayout`).

4. Apply the changes that you just made to the *obj.conf* file on the web server's administration page.
5. Stop and restart the web server.

2.4.4 Updating the SSUrlMapPlugin File

The SSUrlMapPlugin is a web filter plug-in used for mapping Site Studio URLs. The component automatically updates this file during installation. However, you may need to manually update the file if you are using Microsoft Internet Information Server (IIS), and you are upgrading from an earlier Site Studio version. Microsoft IIS typically locks this file and prevents it from being overwritten.

To manually install the SSUrlMapPlugin file, perform these tasks:

1. Stop the content server and the web server.
2. Copy the *[CS_Dir]\custom\SiteStudio\support\win32\SSUrlMapPlugin.dll* file to the *[CS_Dir]\shared\os\win32\lib* directory.
3. Start the content server and the web server.

Important: The content server must be started before the "World Wide Web Publishing" service.

Note: When installing Site Studio in a clustered environment, you will also need to manually update the SSUrlMapPlugin file on each server in the cluster.

2.4.5 Configuring Zone Fields

When you install the Site Studio component, several metadata fields are added to the content server. Some of these must be configured as zone fields to ensure that they are full-text indexed.

Make sure that the DBSearchContainsOpSupport component is installed and enabled on the content server. This component ensures that the zone fields are full-text indexed properly, which is required for Site Studio sites to work correctly. If this component is not installed and enabled, you should install and enable it. You can find the component zip file in the `\packages\allplatform` directory of the Content Server distribution package. See the Content Server installation documentation for more information and instructions.

Configuring Site Studio Metadata Fields as Zone Fields

To configure Site Studio metadata fields as zone fields, perform these tasks:

1. Log in to Content Server as an administrator.
2. Open the content server **Administration** page.
3. Click **Zone Fields Configuration**.
4. Designate **Web Sites** and **Exclude From Lists** as zone text fields.

You do not need to rebuild the search index after enabling these fields as zone fields.

Adding a Setting to the Content Server Configuration File

To add a setting to the Content Server configuration file, perform these tasks:

1. Open a new browser window and log in to Content Server as a system administrator (with the "sysmanager" role).
2. Go to the Administration Applets page and click the **Admin Server** link.
3. On the Content Admin Server page, click the button of the appropriate content server instance.

The status page for the selected content server instance is displayed.

4. In the option list for the server instance, click the **General Configuration** link.

The General Configuration page is displayed.

5. Scroll down to the bottom of the page, and add the following line to the Additional Configuration Variables box:

```
SSUseUniversalQueryFormat=1
```

6. Optionally, you can also add the following line:

```
SSEnableDBSearchShortcut=1
```

This improves the responsiveness of queries performed by Site Studio Designer and Contributor.

7. Click **Save**.
8. Restart the content server.

2.4.6 Enabling JavaServer Pages on the Content Server

If you plan to use JavaServer Pages in Site Studio, you must enable JSP on the content server. This enables you to access and modify content and services (personalization,

security definitions, predefined variables, and so on) on the content server. For more information on enabling JavaServer Pages, see the *Getting Started With the Software Developer's Kit (SDK)* guide, which is part of the Content Server documentation set.

If you enable any JSP groups after you enable the Site Studio component, you must configure the JSP support so the JSP fragments function properly.

Important: JSP is supported only in legacy Site Studio projects; that is, projects that use the pre-10gR4 architecture. These are typically projects that were created in a Site Studio release before 10gR4 and that are opened in Designer 10gR4.

Configuring JSP Support for a New JSP Group

If you install the Site Studio component and then add a group to the list of JSP Enabled Groups in the content server, you must redeploy the JSP support files for that group to allow Site Studio JSP fragments to work correctly.

To configure JSP support, perform these tasks:

1. Log onto Content Server as an administrator.
2. Under Administration, click **Site Studio Administration**.
3. Click **Manage Fragment Libraries**.
4. Click the **Configure JSP Support** button.

The JSP support files are extracted to the required directories on the content server.

2.4.7 Configuring Active Server Pages on the Content Server

If you plan to create ASP sites in Site Studio, you must do the following after installing the StudioStudio component:

Important: ASP is supported only in legacy Site Studio projects; that is, projects that use the pre-10gR4 architecture. These are typically projects that were created in a Site Studio release before 10gR4 and that are opened in Designer 10gR4.

- Install the IdcCommandUX component (version 7.0.0.7 or higher) on the content server. You can find the component zip file in the `\extras` directory of the Content Server software distribution package.
- If you are using Microsoft Windows Server 2003, enable ASP support and configure the server so that it allows Site Studio to use parent paths such as `..\..\websites\`. (For more information, see the documentation related to "parent paths" in the Microsoft Internet Information Server Help.)
- Enable the "Websites" folder in Microsoft Internet Information Server (IIS) to execute scripts and to create an application object. To do this, perform these tasks:
 1. From **Control Panel**, open **Administrative Tools** and then **Internet Information Services**.
 2. Right-click the **Websites** folder and choose **Properties**.
 3. In the Websites Properties dialog box, on the Home Directory tab, select **Scripts only** from the **Execute Permissions** list.

4. In the **Application name** text box, type `websites`.
5. Click **Apply** and then click **OK**.
6. Restart IIS.

2.4.8 Configuring Active Server Pages in a Clustered Content Server Environment

In a non-clustered content server installation, the ASP code identifies the content server process it communicates with through variables in the *global.asa* file found in the `\weblayout\websites` folder. In a clustered environment, each web server in the cluster should communicate with the content server process running on that same server and not reroute communication through the load balancers virtual IP address. In this scenario, a shared *global.asa* file is not adequate, and each web server in the cluster must be set up to use its own unique *global.asa* file.

Important: ASP is supported only in legacy Site Studio projects; that is, projects that use the pre-10gR4 architecture. These are typically projects that were created in a Site Studio release before 10gR4 and that are opened in Designer 10gR4.

If you plan to use Active Server Pages for Site Studio sites in a clustered content server environment, you must do the following after you install the component:

1. For each web server in the cluster, perform the following tasks:
 - Install the `IdcCommandUX` component (version 7.0.0.7 or higher) on the content server. You can find the component zip file in the `\extras` directory of the Content Server software distribution package.
 - If you are using Microsoft Windows Server 2003, enable ASP support on each of the web servers in the cluster and configure the server so that it allows Site Studio to use relative paths such as `..\..\websites\`. (For more information, see the documentation related to "parent paths" in the Microsoft Internet Information Server Help.)
 - Remove any application object that has been created for either the "Websites" or the root weblayout (for example, "idcm1") folder in Microsoft Internet Information Server (IIS) and leave only the root web servers application object.
2. For each web server in the cluster, update the *global.asa* file found in the root web server directory (usually `c:\inetpub\wwwroot`). Create the file, if necessary, and ensure that it contains the following method:

```
<SCRIPT LANGUAGE=vbscript runat=server>
Sub Application_OnStart
Application("ssIdcReference") = "socket:SERVER:4444"
Application("ssIdcWebRoot") = "http://SERVER/idcm1/"
End Sub
</SCRIPT>
```

In the above script, for each web server in the cluster:

- What you replace the word `SERVER` with depends on whether you specified a value for `ClusterNodeAddress` in the installation process for your cluster nodes.
- If you do have a `ClusterNodeAddress` value, then that same value is necessary to replace the word `SERVER`.

- If you do not have a value for `ClusterNodeAddress`, then you can replace the word `server` with any value that identifies the node, typically `127.0.0.1` or `localhost`, although you can also use the name of the system or its IP address.
- Example: If you set `ClusterNodeAddress=10.20.30.40` (And you are using default port `4444` and your relative web root is `'idcm1'`)

Then your *global.asa* file would be:

```
<SCRIPT LANGUAGE=vbscript runat=server>
Sub Application_OnStart
Application("ssIdcReference") = "socket:10.20.30.40:4444"
Application("ssIdcWebRoot") = "http://10.20.30.40/idcm1/"
End Sub
</SCRIPT>
```

- If you do not have `ClusterNodeAddress` set, then your *global.asa* might look like:

```
<SCRIPT LANGUAGE=vbscript runat=server>
Sub Application_OnStart
Application("ssIdcReference") = "socket:127.0.0.1:4444"
Application("ssIdcWebRoot") = "http://127.0.0.1/idcm1/"
End Sub </SCRIPT>
```

- Replace `"idcm1"` with your instance name, as necessary.
- Replace `"4444"` with your specific port number, as necessary.

3. Restart IIS.

2.5 Rebuilding the Content Server Index

If you are using database search and indexing (full-text or metadata-only), you do not need to rebuild the search index when installing or upgrading the Site Studio component on the content server. If you are using a different search engine (typically, Verity), you must rebuild the search index when installing or upgrading the Site Studio component. You should rebuild the search index after enabling the component and configuring the content server.

The index rebuild is necessary to take advantage of new metadata fields introduced by Site Studio.

Important: Rebuilding the search index may be a very time-consuming process, depending on the number of content items managed by your Content Server instance. It is therefore recommended that you perform this rebuild during off-peak hours of Content Server use (typically at night or on the weekend).

If you plan to upgrade Web sites created in Site Studio releases before 7.5 (see [Appendix B, "Upgrading Pre-7.5 Web Sites"](#)), you must rebuild the search index on the content server at that time. Therefore, to prevent rebuilding the index more than once, you may want to skip this step until after you have successfully upgraded your sites.

See the Content Server administration documentation for more information on rebuilding the index.

2.6 Restarting the Web Server

Make sure that you restart the web server after setting up Site Studio. If you do not, you will not be able to access any Site Studio Web sites.

Installing Designer

Site Studio Designer is an application that provides a development environment in which users can design, build, and manage Web sites.

This section covers the following topics:

- ["About Designer"](#) on page 3-1
- ["System Requirements"](#) on page 3-1
- ["Installing Designer"](#) on page 3-2
- ["Launching Designer"](#) on page 3-2

3.1 About Designer

Site Studio Designer provides a development environment in which a single individual (the site designer) can create, design, manage, and distribute Site Studio Web sites. The designer is typically a webmaster, a web developer, a site administrator, or someone in a similar position.

Once a site has been created in Designer, it can be handed off to site managers and contributors, who can create and maintain the Web site content. Typically, one designer works with multiple managers and contributors.

3.2 System Requirements

The system requirements for the Site Studio Designer application are as follows:

- Windows XP (SP2 or higher), Windows Vista (SP1 or higher), Windows 2003 Server (SP2 or higher), Windows 2003 Server R2, Windows Server 2008 (SP1 or higher), Windows 7 operating systems.
- Access to an Oracle Content Server instance running the Site Studio component.
- Microsoft Internet Explorer 5.5 or higher. (You can use Microsoft Internet Explorer 5.5 or higher, or Mozilla Firefox 1.0.7 or higher to view the resulting web pages.)

3.3 Installing Designer

To install the Site Studio Designer application, perform these tasks:

Note: You need administrative rights to the computer to install Site Studio Designer. If you want to install Site Studio Designer on a system running Windows Vista with User Account Control (UAC) turned off, make sure that you install the application as a user with administrator privileges.

1. On the computer that will be used to create and manage Web sites, open the Site Studio software distribution package.
2. Open the **Designer** folder.
3. Launch **setup.exe** and follow the instructions on the screen.

Note: If you install Site Studio Designer 10gR4 on a computer that already has an earlier Designer version on it, the Designer 10gR4 application installs alongside the earlier version. (The previous version is not removed.)

3.4 Launching Designer

After you have installed the Site Studio Designer software, you can open the application from the **Start** button by choosing **Programs**, then **Oracle Universal Content Management**, then **Site Studio 10gR4**, and then **Site Studio Designer**.

The first time Designer opens, it displays an empty site workspace, with the Site Connection Manager dialog box already open (so that you can set up a connection to your first Web site). Thereafter, Designer opens the last Web site you worked on (you can change this setting, if you like).

Note: For more information about using Site Studio Designer, see the *Site Studio Designer Guide*.

Setting Up Contributor

Site Studio Contributor is an application that provides an in-context editing environment for contributors to add and edit content on a Site Studio Web site.

This section covers the following topics:

- ["About Contributor"](#) on page 4-1
- ["System Requirements"](#) on page 4-1
- ["Launching Contributor"](#) on page 4-1

4.1 About Contributor

The site designer can make certain areas on a Web site available to site contributors (the contribution regions). This enables them to edit or update the content in these contribution regions. Site Studio Contributor provides numerous options for editing the web pages of your Web site. It offers an editing environment that looks a lot like a word-processing application and provides most of the commonly used functions from desktop editors. You can write text, format it, add images, create tables, and much more.

4.2 System Requirements

The system requirements for the Site Studio Contributor application are as follows:

- An operating system that can run Internet Explorer 6.0 or higher, or Firefox 2.0 or higher (for example, Microsoft Windows 2000, Windows XP, Windows Vista, Windows 7, Linux, or Mac OS X).
- Access to an Oracle Content Server instance running the Site Studio component.
- Microsoft Internet Explorer 6.0 or higher, or Firefox 2.0 or higher. To view the resulting web pages, you need Microsoft Internet Explorer 5.5 or higher, or Mozilla Firefox 1.0.7 or higher.

4.3 Launching Contributor

The Contributor application opens automatically whenever you edit the content in a contribution region on a web page. As such, you never actually launch Contributor by itself (for example, from your desktop or Start menu), as you would most applications.

The first step for contributors to edit or update content on a web page is to open that page in contribution mode. This is done by pressing a special key combination while viewing the web page in a web browser. The default key combination is Ctrl+Shift+F5, but the site administrator may have set up a different combination.

When you press the assigned key combination, you may be prompted for your login credentials (user name and password) to connect to the content server that the Web site resides on. After you log on, the page is shown in contribution mode, with one or more contribution graphics close to each of the contribution regions (that is, the areas on the web page that can be edited) (Figure 4–1).

Figure 4–1 Contribution Graphic



The contribution graphic shows (from left to right) the name of the contribution region, an edit icon, and a menu icon. If you hover your mouse cursor over a contribution graphic, its associated contribution region is marked in a yellow box. This is the content that can be edited using the selected contribution graphic.

To edit the selected contribution region, click its edit icon, or click the menu icon and choose the **Edit** option. The Contributor editor then opens in a new browser window, and you can start editing the content in that region. Please note that when the Contributor editor opens, the web browser displaying the original web page becomes temporarily unavailable. You can return to the web browser when you close the Contributor editor.

Once you are done editing and save the contribution data file, the Web site is automatically updated to reflect the changes.

For more information about using Site Studio Contributor, see the *Site Studio Contributor Guide*.

Installation Considerations

This section covers the following installation considerations:

- ["Site Studio Metadata Fields"](#) on page 5-1
- ["Changing the Default Contributor Keystroke Combination"](#) on page 5-4
- ["Blocking Access to the Consumption Server"](#) on page 5-5
- ["Changing the Default Contributor Editor"](#) on page 5-5
- ["Updating Custom Elements"](#) on page 5-5
- ["Using the Certificate Validation Applet"](#) on page 5-6
- ["Configuring a Single Sign-On \(SSO\) Environment"](#) on page 5-7
- ["Enabling Debugging in Contributor"](#) on page 5-8
- ["Using Pre-10gR4 Site Studio Projects"](#) on page 5-8

5.1 Site Studio Metadata Fields

When you install the Site Studio component, five metadata fields are added to the content server:

- [Web Sites](#)
- [Web Site Section](#)
- [Web Site Object Type](#)
- [Exclude From Lists](#)
- [Region Definition](#)

5.1.1 Web Sites

The Web Sites metadata field (actual name is "xWebsites") contains a list of Web sites that a particular site asset on the content server is associated with. Each Web site has its own ID, and each time you add a new asset or content file to a Web site, the Web site ID is automatically added to this metadata field for that file.

The Web Sites metadata is needed for dynamic lists and searches to function properly. You can also use this metadata to quickly view all of the Web sites that a particular file is associated with.

The Web Sites metadata was introduced to accommodate content reuse. It replaces the WebsiteID metadata (a deprecated field that provides support for older versions of Site Studio fragments). If you are upgrading from an earlier version of Site Studio, you

should customize your fragments that currently use the WebsiteID value, even if you do not intend to take advantage of content reuse. For more information, see ["Performing Additional Steps Manually"](#) on page B-6.

5.1.2 Web Site Section

The Web Site Section metadata field (actual name is "xWebsiteSection") is populated automatically when you install the new Site Studio component on the server. The Web site section indicates where a document is stored on a Web site (unless a target section is explicitly specified in the original hyperlink).

For existing content on a Web site (created with an earlier version of Site Studio), the Web Site Section value is derived from the "Collection ID" value. (Collection IDs were a part of the Folders component, which is no longer used by Site Studio.) All of this is handled automatically when you upgrade a site. For more information, see ["Upgrading Sites on a Single Content Server Instance"](#) on page B-3.

When you complete the site upgrade, you will want to start using Web site sections instead of folders. If you want to continue using folders, see the steps in ["Assigning a Web Site Section to Your Folders"](#) on page B-10.

5.1.3 Web Site Object Type

The Web Site Object Type metadata field (actual name is "xWebsiteObjectType") is added automatically when you enable the Site Studio component on the content server. Each metadata value represents a type of site asset that you can use in Site Studio:

- **Data File:** Content files in XML format that are generated by Site Studio. Contributor data files are edited using the Site Studio Contributor application.
- **Native Document:** Content files created using familiar third-party applications such as Microsoft Word. Native documents are converted to HTML format using Dynamic Converter, and they are edited using their associated application.
- **Fragment:** Chunks of code that enhance the functionality of a Site Studio Web site (for example, by providing dynamic navigation aids or a standard page footer).
- **Image:** Graphic files (JPG, GIF, PNG) that are included in content files or page templates (for example, corporate banners or product images).
- **Script:** JavaScript files that provide a series of commands that can be executed without user interaction. Scripts are often used to provide additional functionality to web pages.
- **Stylesheet:** Cascading style sheet (CSS) files that provide control over how page content is displayed (more specifically, how different HTML elements, such as headers and links, appear on the page). Links to CSS files are often embedded in page templates, so their formatting rules apply to all web pages based on these templates.
- **Project:** XML files that store all information about a Site Studio Web site that Designer needs to work with the site, such as the site hierarchy, site section properties, data associations, placeholder mappings, and so on
- **Custom Element Form:** HTML files that define custom forms for use in elements (for example, selection forms for specific file types). Site Studio comes with several predefined custom element forms (in `[CS-Dir]\custom\SiteStudio\elementforms`).

- **Custom Configuration Script:** JavaScript files that override the default Contributor editor configuration to provide contributors with a customized editing experience.
- **Validation Script:** JavaScript files that define the validation rules for element data to determine that the data meets the requirements (for example, it does not exceed a certain maximum length or contain some illegal characters).
- **Manager Settings:** Files that define the functionality that is available in Site Studio Manager. Manager is a web-based tool that allows designated users (site managers) to modify the structure of a Web site.
- **Page Template:** Fully-formed HTML files that define the layout and high-level look-and-feel of web pages, including the placement of contribution regions (that is, editable areas on the page), navigation aids (in the form of fragments) and site-wide images (banners and the like). Page templates are the highest-level site design object.
- **Subtemplate:** Partial HTML files (that is, without head and body sections) that can be inserted into placeholders on page templates to divide them into further smaller, reusable areas with their own placeholders and contribution regions.
- **Region Template:** Partial HTML files (that is, without head and body sections) that define the layout and look-and-feel of the data in contribution regions within web pages.
- **Placeholder Definition:** Files that define what region definitions, region templates, and subtemplates are allowed for the associated placeholders. They also specify what contributor actions are allowed for the placeholders.
- **Region Definition:** Files that define the type of content that elements of a particular type consists of. They also specify the content creation and switching options available to contributors for contribution regions, and set default metadata for content files associated with these regions.
- **Element Definition:** Files that define the editing experience for element types. Specifically, they specify what a contributor can do when editing an element.
- **Conversions Definition:** Files that specify the conversion rules for native documents on a Web site.
- **Other:** Any other media files that could be used on a Web site, such as Flash animations, video files, audio files, and so on.

See the *Site Studio Designer Guide* for detailed explanations of each of these file types.

5.1.4 Exclude From Lists

The Exclude From Lists metadata field (actual name is "xDontShowInListsforWebsites") lists the Web sites for which a contributor has specified, through the user interface, that a contributor data file or native document should *not* display in dynamic lists on the Web site.

When a contributor excludes a file from a dynamic list, the Web site ID is added to this value. If the contributor later re-includes the content in dynamic lists for the Web site, the Web site ID is removed from this metadata field and the content becomes available to the dynamic lists again.

Note: If a Web site value displays in this metadata field for a particular data file or native document, then that piece of content will not display in any lists on the site; however, it will still display in search results for the site.

5.1.5 Region Definition

The Region Definition metadata field (actual name is "xRegionDefinition") specifies the region definition that a contributor data file is associated with. A data file can be associated with only one region definition, but a region definition may be associated with many data files.

Region definitions define the types of content used on a Web site. They could be thought of as 'content classes'. They are essentially groups of individual elements which define the various chunks of reusable information for a particular site content type. For example, there could be a region definition ('content class') called "Press-Release," which consists of the elements Title, Subtitle, Intro-Text, Body-Text, and Image. Contributor data files are associated with a region definition to store the data for each element in the region definition. (What a contributor can do with the data is controlled by element definitions.)

In addition to defining site content types in terms of its constituent parts (elements), region definitions also specify the content creation and switching options available to contributors for its associated contribution region(s). For example, if a contribution region is set up to allow contributors to switch the content of that region, they might be allowed to use existing contributor data files on the server only (not native documents or new contributor data files). (Please note that placeholder definitions control whether contributors can actually switch content in contribution regions.) Finally, region definitions also set the default metadata for content in contribution regions as it is checked into the content server.

See the *Site Studio Designer Guide* for more information about region definitions.

5.2 Changing the Default Contributor Keystroke Combination

The default keystroke combination for entering contribution mode on a web page is Ctrl+Shift+F5, but you can change this. To do so, you need access to the "custom" directory on the content server where Site Studio is installed. You should inform the designers and contributors of the site if you change this value.

To change the default keystroke combination, perform these tasks:

1. Browse to the following directory (where *[CS-Dir]* is the installation location of your content server):

```
[CS-Dir]\custom\SiteStudio\publish\resources\wcm\sitestudio\
```

2. Open **wcm.toggle.js** in a text editor.
3. Locate the function `OnKeyDown`.
4. Change the implementation of this function to use a different keystroke combination that will call `WCM.CONTRIBUTOR.Toggle`.

This function uses virtual key codes to determine the key combination entered by the user. The default value is Ctrl+Shift+F5. The F5 key has a virtual key code of 116 (or 0x74 in hexadecimal). The codes for the other typical function keys, F1 through F12 are 112 (0x70) through 123 (0x7B), respectively.

5. Save and close `wcm.toggle.js`.

Note: The next time you upgrade Site Studio or install a patch, you may need to perform these steps again to retain your keystroke combination.

Note: The key codes used to determine the keystrokes should be given special consideration in instances where contributors may use different operating systems, since the virtual key codes may vary among operating systems.

5.3 Blocking Access to the Consumption Server

A contributor gains access to the content of a Web site by using a keystroke combination to enter contribution mode and then clicking a contribution icon to launch the Contributor application.

While such access is necessary on the server used for building the Web site, it is not desirable for the consumption server; that is, the server used to run the live Web site. To block contributor access to the consumption server, create the following server configuration variable (in `[CS-Dir]/config/config.cfg`):

```
DisableSiteStudioContribution=true
```

If the variable is not present or is set to "false," contributor access is allowed.

Make sure that you restart the content server.

5.4 Changing the Default Contributor Editor

The default editor used for the Contributor application is FCKeditor, but this can be changed to another editor, such as Ephox.

If you want to use Ephox as the contribution editor, include the following server configuration variable (in `[CS-Dir]/config/config.cfg`):

```
SSDefaultEditor=ephox
```

If you want to change the default editor back to FCKeditor, remove this variable or change it to:

```
SSDefaultEditor=fck
```

Make sure that you restart the content server.

5.5 Updating Custom Elements

Any custom element forms created using a Site Studio release before 10gR3 (10.1.3.3.3) are not compatible with Site Studio 10gR4. They must be manually upgraded and re-authored. The primary reason for not maintaining backward compatibility is Site Studio's prior dependency on Internet Explorer's proprietary 'window.external' functionality (due to the ActiveX control used for the Contributor application). This functionality was removed from Site Studio as a result of the browser-independent, JavaScript-based Contributor application that is used in Site Studio 10gR3 (10.1.3.3.3)

and higher. See the technical reference guide for Site Studio 10gR3 for more information.

5.6 Using the Certificate Validation Applet

If you set up Site Studio to use Ephox as the contributor editor, the Contributor application uses a signed Java applet that has one or more signed plug-ins packaged with it. The first time that these applets are loaded on any given system, the user is prompted to accept the security certificates. Some Java Virtual Machines (JVMs) have a problem when multiple signed applets attempt to load at the same time, which may cause the browser to hang. If you encounter this problem on any client computers, there are two solutions. One is to have the security certificates pushed out to the client computers by your IT department. The other solution is to manually accept the certificates one at a time. Site Studio provides a Site Studio Certificate Validation page on the content server, where users can accept the certificates. This page is accessed from the user profile page (under "My Profile").

Since it is possible to build custom plug-ins for Site Studio Contributor, this page has been made extensible to allow the certificates to be accepted in the same way for any custom signed plug-ins. The following server configuration variables are used in this context (in `[CS-Dir]/config/config.cfg`):

- `SSEExtraCertificateClasses`
- `SSEExtraCertificateLabels`
- `SSEExtraCertificateJars`
- `SSHttLayerManager`

SSEExtraCertificateClasses

This entry appends customer-specific entries to the list of classes loaded by the certificate validation applet. The value is a space-separated list of classes to load during the certificate validation process, for example:

```
SSEExtraCertificateClasses=com.xalco.XalcoEphoxPlugin com.zeng.TextGenerator
```

Notes:

- Each class in the list should have a corresponding label set using the `SSEExtraCertificateLabels` entry.
- This should be used with the `SSEExtraCertificateLabels` and `SSEExtraCertificateJars` entries.

SSEExtraCertificateLabels

This entry appends customer-specific labels to the list of certificates checked by the certificate validation applet. The value is a caret-separated list of certificate descriptions to display during the certificate validation process, for example:

```
SSEExtraCertificateLabels=Xalco Certificate^Ravenna Certificate
```

Notes:

- Each label in the list should have a corresponding class set using the `SSEExtraCertificateClasses` entry.
- This should be used with the `SSEExtraCertificateClasses` and `SSEExtraCertificateJars` entries.

SSEExtraCertificateJars

This entry appends customer-specific entries to the classpath used by the certificate validation applet. This allows the JVM to find the classes listed in the `SSEExtraCertificateClasses` entry. The value is a comma-delimited list of certificate descriptions to display during the certificate validation process, for example:

```
SSEExtraCertificateJars=<$HttpWebRoot$>resources/xalco/XalcoEphoxPlugin.jar,
<$HttpWebRoot$>groups/public/documents/adacct/HelloWorldPlugin.jar
```

Notes:

- Idoc Script tags embedded in the entries are evaluated.
- This should be used with the `SSEExtraCertificateClasses` and `SSEExtraCertificateLabels` entries.

SSHttpLayerManager

This entry allows the use of the "Sun" or default `HttpLayer` manager inside the Ephox Editor. Changing to the Sun layer may provide better results when running in an SSL environment. The possible values are:

- `SSHttpLayerManager=default`: Uses the default, internal Ephox `HttpLayer` manager.
- `SSHttpLayerManager=sun`: Uses the "Sun" `HttpLayer` manager.

This corresponds to Ephox's `setHttpLayerManager` configuration entry. See http://www.ephox.com/developers/editliveforjava/v50/html/prop_httpmanagerlayer.html for more information.

5.7 Configuring a Single Sign-On (SSO) Environment

Site Studio 10gR4 can be used in forms-based authentication and single sign-on (SSO) environments. The following HTML comment must be added to the login page that prompts the user for credentials:

```
<!--IdcClientLoginForm=1-->
```

This token must be used verbatim, without any spaces or case changes. Please note that if the login form does not contain this HTML comment, Site Studio Designer cannot connect successfully when attempting to connect to a Web site protected by a forms-based login. Site Studio Designer then displays a "200 OK" message, but fails to connect.

In addition, if the forms-based login solution delivers the login form to the client without using a redirect, as is the case with Content Server's ExtranetLook component, the following server configuration variables must be added to `[CS-Dir]/config/config.cfg`:

```
SSEnableExtranetLookCompatibility=1
```

Do not forget to restart the content server after modifying the `config.cfg` file. If you do not add this configuration variable, the browser shows a CGI-based URL after entering contribution mode in a Web site instead of a path-based URL.

5.8 Enabling Debugging in Contributor

Site Studio Contributor provides a console window as the application's x-browser/platform logging mechanism. This console window is especially suited to accommodate logging and to facilitate JavaScript code execution across multiple contexts (HTML window objects) from within a single browser window. All instructions logged to the console window's logging window displays the time of execution and the context in which the instruction was executed.

The availability of the console window is optional and is not deployed to the web-accessible directory of the content server ('weblayout') by default. A *wcm.console.htm* file is provided in the *[CS-Dir]\custom\SiteStudio\support* directory. To enable the console window, this file should be copied to the 'base' directory within the content server's web-accessible directory: *[CS-Dir]\weblayout\resources\wcm\base*.

With the *wcm.console.htm* file in place, there are two main ways to launch the console window:

- **Using the console window's URL:** Users can navigate directly to the Console Window if the URL is known. The following is where the console window usually exists in a Site Studio installation: `http://[server]/[instance]/resources/wcm/base/wcm.console.htm`.
- **Use key commands:** From within the Contributor application, the key command `Ctrl+Alt+Shift+C` or `Ctrl+Alt+Shift+E` launches the console window if it is not already open. The latter key command displays all the run-time errors in addition to opening the console window. If the console window is not installed in the content server's web-accessible directory ('weblayout'), then pressing `Ctrl+Alt+Shift+E` shows the generic error dialog if there are any run-time errors. The error dialog only displays run-time errors, not run-time logging.

Debugging Logging Level (Ephox)

If Site Studio has been set up to use Ephox as the contributor editor, you can use the `SSEditorDebugLevel` server configuration variable (in *[CS-Dir]/config/config.cfg*) to control the amount of logging that the editor emits to its Java console. This entry can be set to one of the following values:

- fatal
- error
- warn
- info
- debug
- http

The default value of the setting is "info." Do not forget to restart the content server after modifying the *config.cfg* file.

5.9 Using Pre-10gR4 Site Studio Projects

Site Studio 10gR4 is backward compatible with earlier releases. This means that you can use Site Studio Designer 10gR4 to work with projects created in earlier Site Studio releases. You do not need to upgrade them per se. It is important to note, though, that these projects will continue to work in "legacy" mode; that is, they will use the pre-10gR4 architecture and they will not take advantage of the new architecture and features introduced in Site Studio 10gR4.

If you want to use Site Studio projects created in Designer releases before 7.5, you must upgrade them first. See [Appendix B, "Upgrading Pre-7.5 Web Sites"](#) for more information. Please note that, after upgrading, these projects will work as "legacy" projects.

Uninstalling Site Studio Software

This section covers the following topics:

- ["Uninstalling Designer"](#) on page A-1
- ["Uninstalling the Site Studio Component"](#) on page A-1

A.1 Uninstalling Designer

To uninstall Site Studio Designer from a client computer, perform the following tasks:

1. Open Windows Control Panel.
2. Click the following link, depending on your operating system:
 - Windows XP: **Add or Remove Programs**
 - Windows Vista and Windows 7 (category view): **Uninstall a program** (under Programs)
 - Windows Vista and Windows 7 (icon view): **Programs and Features**
3. Select **Site Studio Designer 10gR4**.
4. Do the following, depending on your operating system:
 - Windows XP: Click **Remove**.
 - Windows Vista and Windows 7: Right-click and choose **Uninstall**.

The Desktop Integration Suite uninstall wizard is launched.
5. Click **Yes** when asked if you are sure that you want to remove the Site Studio Designer software from your computer.

The software is removed from the client computer.

A.2 Uninstalling the Site Studio Component

You may want to disable (or uninstall) the Site Studio component. Disabling a component instead of uninstalling it leaves open the option of enabling it again later, without the need to reinstall.

To disable or uninstall the Site Studio component, perform these tasks:

1. Open a new browser window and log in to Content Server as a system administrator (with the "sysmanager" role).
2. Go to the Administration Applets page and click the **Admin Server** link.

3. On the Content Admin Server page, click the button of the content server instance that you want to uninstall the Site Studio component from.

The status page for the content server instance is displayed.

4. In the option list for the server instance, click the **Component Manager** link.

The Component Manager page is displayed.

5. Under Enabled Components, select **SiteStudio**.

6. Click **Disable**.

7. Restart the content server.

8. If you want to fully uninstall the component rather than simply disabling it, return to the Component Manager page, select **SiteStudio** from the Uninstall Component list, and click **Uninstall**. Then restart the content server.

Upgrading Pre-7.5 Web Sites

This section covers the following topics:

- ["Introduction"](#) on page B-1
- ["What the Automated Upgrade Does"](#) on page B-1
- ["Upgrading Your Content Servers"](#) on page B-2
- ["Performing Additional Steps Manually"](#) on page B-6

B.1 Introduction

If you are upgrading from a Site Studio release before 7.5, you must upgrade your Web sites before you can use them with Site Studio 10gR4. This is because Site Studio versions 7.5 and 10gR3 have some important architectural changes, including:

- The site hierarchy is stored in a project file and no longer relies on folders. As a result, Content Server's folders features are no longer required.
- The Web site URLs display as a logical path and suffix instead of as a CGI-based address displaying the SS_GET_PAGE service. As a result, you see friendlier, path-based URLs.
- Layout pages no longer use the `<base>` tag. Therefore, hyperlinks and references that rely on the base tag must be modified.
- The "siteId" and "root node id" are no longer synonymous.

It is important to realize that, after upgrading, the upgraded projects will work as "legacy" projects in Site Studio 10gR4; that is, they will use the pre-10gR4 architecture and they will not take advantage of the new architecture and features introduced in Site Studio 10gR4. The same is true of projects that were created using Site Studio 7.5 or 10gR3. They do not need to be upgraded per se and you can use them with Site Studio 10gR4, but they will continue to function in "legacy" (i.e., pre-10gR4) mode.

B.2 What the Automated Upgrade Does

When you upgrade your pre-7.5 Site Studio release and your Web sites, the following tasks are performed automatically:

Action	Description
Folders-based sites upgraded to project-based sites	The existing hierarchy in the folder structure is reproduced in the project file. The root "dCollectionName" is used as the "siteLabel," the root "dCollectionID" is used as the "siteId," the "originalCollectionID" project attribute is set, and the site type is transferred from the root section to the project.
Custom section properties in new sites updated	The custom section properties of type "siteid" and "url" are updated (adding friendly URLs, where necessary).
Fragment instance parameters in layout pages updated	Parameters of type "managedurl" and "url" are updated.
Metadata populated	If the Create Project Files option is enabled, the "xWebsiteSection" values are populated (derived from "xCollectionID").
Links in layout pages and data files updated	If the Upgrade Layouts and Upgrade Data Files options are enabled, the weblayout links in layout pages and contributor data files are updated to include the HttpRelativeWebRoot token; optionally, the javascript links are updated.
Navigation updated	The navigation files for the Web site are regenerated.

Note: Custom elements cannot be upgraded automatically. See ["Updating Your Custom Elements"](#) on page B-10 for more information.

B.3 Upgrading Your Content Servers

The task of site upgrade begins with upgrading the Site Studio component on each of the content servers you are using, and then upgrading the Web sites stored on the content servers:

- ["Upgrading Sites on a Single Content Server Instance"](#) on page B-3
- ["Upgrading Sites on Multiple Content Server Instances"](#) on page B-3
- ["Performing a Full Upgrade"](#) on page B-4
- ["Performing a Minimal Upgrade"](#) on page B-5

Although the Folders component is not used in Site Studio version 7.5 or later, you must retain folders during the upgrade of your Web sites so that each site can be migrated from a folders-based hierarchy to a project-based hierarchy.

You can then disable the Folders component (assuming you are not using the Check Out and Open component, which requires folders). If you want to continue using folders, you must configure them with the appropriate metadata (see ["Assigning a Web Site Section to Your Folders"](#) on page B-10).

Note: When you follow the upgrade steps, every Web site on the server is upgraded. If you want to upgrade only selected sites, then you must create a copy of the other sites on another server.

B.3.1 Upgrading Sites on a Single Content Server Instance

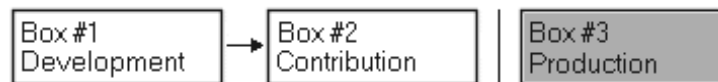
If your Web sites are stored on a single content server, the upgrade consists of:

1. Installing the new Site Studio component (having first uninstalled the old component). For more information, see [Chapter 2, "Installing the Site Studio Component."](#)
2. Performing a full upgrade on the content server. For more information, see ["Performing a Full Upgrade"](#) on page B-4.
3. Performing additional steps manually that are not handled by the automated upgrade. For more information, see ["Performing Additional Steps Manually"](#) on page B-6.

B.3.2 Upgrading Sites on Multiple Content Server Instances

You may have sites on multiple content servers, each serving a different purpose, such as a development server, a contribution server, and a production server.

Figure B-1 Multiple Content Server Instance Illustration



The content on each server (source server) gets copied to the next server (target server) using the Archiver/Replicator utility. As such, it is important to carefully plan and upgrade the sites on each server without encountering replication problems.

On the First Two Instances of the Content Server

- Stop replication between the content servers
- Install the new Site Studio component. For more information, see [Chapter 2, "Installing the Site Studio Component."](#)

On the Source Instance of the Content Server

- Perform a full upgrade of your sites. For more information, see ["Performing a Full Upgrade"](#) on page B-4. Then...
- Perform additional steps manually (steps not handled by the automatic upgrade). For more information, see ["Performing Additional Steps Manually"](#) on page B-6.

On the Target Instance of the Content Server

- Perform a minimal upgrade of your sites. For more information, see ["Performing a Minimal Upgrade"](#) on page B-5.

On Both Instances of the Content Server

- Start replication again between the content servers.

Once the new component has been installed on all instances of the content server and Web sites have been upgraded as indicated above, you can begin replicating your sites again.

You can use the replication feature in Site Studio (see the *Site Studio Administrator and Manager Guide*). Or, if you have been using Archiver/Replicator and want to continue using it, you can do so as long as you modify the archive query to include Site Studio project files.

On the Next Target Content Server (Downstream in Replication)

- Stop replication between the source and target content servers

Note: In this case, your source server (Box 2) was the target server in the previous steps, and your target server (Box 3) is the next server down the line (downstream) in your replication.

- Install the new Site Studio component. For more information, see [Chapter 2, "Installing the Site Studio Component."](#) Then...
- Perform a minimal upgrade of your sites. For more information, see ["Performing a Minimal Upgrade"](#) on page B-5. Then...
- Start replication again between the source and target content servers.

Repeat this last procedure for each target instance of the content server downstream in your replication.

B.3.3 Performing a Full Upgrade

A full upgrade of the content server is required in the case of single-server setup. It is also required for the source server in the case of a multi-server setup. (All other servers in a multi-server setup require a minimal upgrade.)

When you upgrade your site, Site Studio turns your existing folder-based site into a project-based site. When it does this, it creates a project file as a managed item in the content server. As such, you must identify the metadata that you would like assigned to the project file that will represent each Web site.

During the upgrade process, the content server attempts to index content that gets changed, which could take considerable time and resources. You may want to temporarily disable automatic indexing before you begin the upgrade process and then re-enable it when you are done. (See the Content Server administration documentation for further details.)

You should have already installed and enabled the new Site Studio component on the server before you start a full upgrade. For more information, see [Chapter 2, "Installing the Site Studio Component."](#)

To perform a full upgrade, perform these tasks:

1. Log on to the content server as an administrator, open the **Administration** page, and then the **Site Studio Administration** page.

2. Click **Set Project Default Document Information**.

The Set Project Default Document Information page is displayed. This is where you assign the default metadata for the new projects that you create in Site Studio.

3. Once you have specified the metadata values, click **Update**.

This returns you to the Site Studio Administration page, where you can begin the upgrade process.

4. Click **Manage Web Sites**.

5. Click **Go to Web Sites Update Page**. (This option displays only when older Web sites are detected.)

6. Click **Advanced Options** to specify site upgrade options.

Figure B–2 Advanced Upgrade Options Screen

7. Choose the following for a full upgrade:
 - Select **Create Project files**.
 - Select **Upgrade Layouts**.
 - Select **Upgrade Data Files**.
 - Select **Convert Hyperlinks** and choose a link format:
 - **To Server-Side Links:** Links contain the coded identity of the target location using server-side script.
 - **To Path-Based URLs:** Links contain the full path to the target location.
8. Click **Set Options** to return to the Upgrade Legacy Web Sites page.
9. Click **Start Upgrade**.

You see the individual files that must be upgraded on this page. Wait until you see the message that says that the upgrade process was completed.

Note: The site upgrade automatically updates the site hierarchy and its many links, and the Web Sites menu in the content server now lists your sites.

B.3.4 Performing a Minimal Upgrade

A minimal upgrade is required in the case of a multi-server setup and applies to all target servers; that is, the server that has Web sites being replicated to it.

You should have already installed and enabled the new Site Studio component on the server before you start a minimal upgrade. For more information, see [Chapter 2, "Installing the Site Studio Component."](#)

To perform a minimal upgrade, perform these tasks:

1. Log onto the content server as an administrator, open the **Administration** page, and then the **Site Studio Administration** page.
2. Click **Set Default Project Document Information**.

The Set Project Default Document Information page is displayed. This is where you assign the default metadata for the new projects that you create in Site Studio.

3. Once you have specified the metadata values, click **Update**.

This returns you to the Site Studio Administration page, where you can begin the upgrade process.

4. Click **Manage Web Sites**.
5. Click **Go to Web Sites Update Page**. (This option displays only when older Web sites are detected.)
6. Click **Advanced Options** to specify site upgrade options ([Figure B-2](#)).
7. Select **Create Project files**.

Note: This upgrades project files and populates the "Web Site Section" metadata value.

8. Click **Set Options** to return to the Upgrade Legacy Web Sites page.
9. Click **Start Upgrade**.
Wait until you see the message that says that the upgrade process was completed. The Web Sites menu in the content server now lists your sites.

B.4 Performing Additional Steps Manually

Once you have upgraded your Web sites, there are still several steps that you must perform manually, including the following:

- ["Updating the Site Navigation"](#) on page B-6
- ["Rebuilding the Content Server Index"](#) on page B-6
- ["Updating Your Custom Fragments"](#) on page B-7
- ["Updating Your Custom Elements"](#) on page B-10
- ["Assigning a Web Site Section to Your Folders"](#) on page B-10
- ["Updating JSP Code"](#) on page B-11

B.4.1 Updating the Site Navigation

After upgrading a pre-7.5 Web site, you must update its navigation files. You can do this in Designer (using the **Update Navigation** button) or on the Site Studio Administration page (specifically, the Manage Web Sites page). This step is necessary for Contributor to function correctly on the site.

B.4.2 Rebuilding the Content Server Index

After upgrading a pre-7.5 Web site, you may need to rebuild the content server index. If your content server has been set up to use database search and indexing (full-text or metadata-only), you do not need to rebuild the search index. If you are using a different search engine (typically, Verity), you must rebuild the search index. This is necessary because Site Studio updates the xWebsiteSection metadata field for all content items residing in folders on your site.

Caution: Rebuilding the search index may be a very time-consuming process, depending on the number of content items managed by your Content Server instance. It is therefore recommended that you perform this rebuild during off-peak hours of Content Server use (typically at night or on the weekend).

See the Content Server administration documentation for more information on rebuilding the index.

B.4.3 Updating Your Custom Fragments

Most of the manual updates you must perform after upgrading your pre-7.5 Web site involve modifying your custom fragments. If you are currently using the predefined fragments that came with Site Studio, you do not have to do this because an updated version of each fragment is included with each Site Studio release.

Most likely, you have customized the fragments or introduced new ones to meet a specific purpose for your organization. There are three things you must do with the fragments so that they work in the latest version:

- [Modifying Links That Rely on the <base> Tag](#)
- [Modifying Obsolete SS_GET_PAGE / JavaScript Links](#)
- [Updating GET_SEARCH_RESULTS](#)

B.4.3.1 Modifying Links That Rely on the <base> Tag

The <base> tag that points to the content server's web-accessible directory ('weblayout') is no longer used. During the site upgrade, Site Studio updates the necessary code in your layout pages and contributor data files, but you must perform this step manually in your custom fragments and scripts.

You can do this by re-authoring hand-coded links that are relative to the URL in the <base> tag and use the `HttpRelativeWebRoot` server-side variable instead.

Example

Say, you have a link to a graphic that looks something like this:

```

```

You must then replace it with the following:

```

```

B.4.3.2 Modifying Obsolete SS_GET_PAGE / JavaScript Links

If any existing fragments use `SS_GET_PAGE`, `javascript:link`, or `javascript:nodelink` style hyperlinks, you may want to change them to path-based URLs to take advantage of their many benefits. (For more information, see the *Site Studio Designer Guide*.)

Example

Say, you have a link that looks something like this:

```
<a href="javascript:nodelink(42);">link</a>
```

You must then replace it with the following:

```
<a href="<!--ssServerRelativeSiteRoot-->products/servers/index.htm">link</a>
```

B.4.3.3 Updating GET_SEARCH_RESULTS

Any fragment that used the `GET_SEARCH_RESULTS` service will continue to work, but will not take advantage of the features provided by Site Studio 7.5 and 10gR3 until it is upgraded to use the `SS_GET_SEARCH_RESULTS` service.

Using the `SS_GET_SEARCH_RESULTS` service has several advantages:

- **limitscope logic** (now provided by the service and not required in the fragment): This limits the search results to only those items within the current Web site.
- **dontshowinlists logic** (now provided by the service and not required in the fragment): This limits the search results to only those items that have not been removed from lists by contributors.
- **ssUrl**: This column provides a friendly URL for each row in the search results.

Fragments that use the GET_SEARCH_RESULTS service are typically dynamic list fragments and search results navigation fragments. The updates required differ depending on the version of the Site Studio release that you are upgrading from:

- If you have been using Site Studio version 6.5, and you have customized dynamic lists or search results fragments using that version (for example, by copying a Site Studio fragment and adding custom code to it), you will have used code that performs limitscope logic using the old xWebsiteID metadata field.
- If you have been using Site Studio version 7.2, and you have customized dynamic lists or search results fragments using that version (for example, by copying a Site Studio fragment and adding custom code to it), you will have used code that performs limitscope logic using the new xWebsites metadata field. In addition, you will have used code that performs 'dontshowinlists' logic using the new xDontShowInListsForWebsites metadata field.

In both cases above, you must update those fragments to remove the old limitscope and dontshowinlists logic from them and to use the new SS_GET_SEARCH_RESULTS service, which now provides this functionality internally.

Example

In Site Studio 6.5, the Standard Dynamic List fragment includes the following code for the SSLimitScope parameter. This should be removed:

```
<!--$QueryText=eval(ssQueryText)-->
<!--$if ssLimitScope like "true"-->
  <!--$if strEquals(QueryText, '')-->
    <!--$QueryText='xWebSiteID=' & siteId-->
  <!--$else-->
    <!--$QueryText=(' & QueryText & ') and (xWebSiteID=' & siteId & )'-->
  <!--$endif-->
<!--$endif-->
```

In Site Studio 7.2, the Standard Dynamic List fragment includes the following code for the SSLimitScope parameter. This should be removed:

```
<!--$QueryText=eval(ssQueryText)-->
<!--$if ssLimitScope like "true"-->
  <!--$if strEquals(QueryText, '')-->
    <!--$QueryText='xWebsites &lt;contains&gt; ' & siteId-->
  <!--$else-->
    <!--$QueryText=(' & QueryText & ') and (xWebsites &lt;contains&gt;' &
siteId & )'-->
  <!--$endif-->
<!--$endif-->

<!--$if strEquals(QueryText, '')-->
  <!--$QueryText='not(xDontShowInListsForWebsites &lt;contains&gt; ' & siteId &
)'-->
<!--$else-->
  <!--$QueryText=(' & QueryText & ') and not(xDontShowInListsForWebsites
&lt;contains&gt; ' & siteId & )'-->
```

```
<!--$endif-->
```

Once the old limitscope logic is removed from the fragment, change the GET_SEARCH_RESULTS service call to use SS_GET_SEARCH_RESULTS. Before you invoke the SS_GET_SEARCH_RESULTS service, however, you should set the following parameter values:

Parameter	Description
ssLimitScope	Specifies that the limitscope logic should be applied by the SS_GET_SEARCH_RESULTS service. Typically this true/false value is supplied by a fragment parameter value.
ssDontShowInLists	Specifies that the dontshowinlists logic should be applied by the SS_GET_SEARCH_RESULTS service. Typically this true/false value is set to "true" in all fragments.
ssTargetNodeId	Specifies the node ID that is used to display the search results. The "ssTargetSiteId" can also be used to generate links to other Web sites on the content server. If the "ssTargetSiteId" is not specified, the generated link assumes the same site that originated the link.
ssTargetSiteId	Specifies the site ID that is used to display the search results. The "ssTargetNodeId" parameter must also be used to fully qualify the target node.
ssSourceNodeId	Indicates the node ID for the current page containing the link.
ssSourceSiteId	Indicates the site ID for the current page containing the link.
ssWebsiteObjectType	Specifies that the search results should be limited to a specific Website Object Type. Typically you leave this value empty.
ssUserSearchText	Specifies any user text to perform a full text search. Typically, this only applies to Search Results fragments where the value is provided by a consumer entering a value in a Search Box fragment.

When looping through the results of the SS_GET_SEARCH_RESULTS service call, you typically use the new "ssUrl" column of the result set if you want to create hyperlinks to that item. This ensures that full path-based URLs are used instead of cryptic ID-based URLs.

Additionally, these URLs should be appended with parameters that describe the source location of the link. This allows error pages to be generated properly when there are invalid links.

The following parameters should be affixed to the URLs.

Parameter	Description
ssSourceNodeId	Declares the source node ID. Used to generate friendly URLs if both ssTargetNodeId and xWebsiteSection are blank.
ssSourceSiteId	Declares the source site ID. This allows the error page to be displayed if the target page cannot be found.

Here is a simplified example using Idoc Script:

```
<!-- New params for SS_GET_SEARCH_RESULTS -->
<!--$ssLimitScope="true"-->
<!--$ssDontShowInLists="true"-->
<!--$ssTargetNodeId=""-->
<!--$ssTargetSiteId=""-->
<!--$ssSourceNodeId=nodeId-->
```

```

<!--$ssSourceSiteId=siteId-->
<!--$ssWebsiteObjectType=""-->
<!--$ssUserText=""-->

<!--$executeService("SS_GET_SEARCH_RESULTS")-->

<!--$loop SearchResults-->
  <a href="<!--$ssUrl-->?ssSourceSiteId=<!--$siteId-->&ssSourceNodeId=
<!--nodeId-->">
    <!--$dDocTitle-->
  </a><br><br>
<!--$endloop-->

```

For more details, refer to the dynamic list and search results fragments that are provided with the Site Studio product.

B.4.4 Updating Your Custom Elements

Any custom element forms created using a Site Studio release before 7.5 are not compatible with Site Studio 10gR4. They must be manually upgraded and re-authored. The primary reason for not maintaining backward compatibility is Site Studio's prior dependency on Internet Explorer's proprietary 'window.external' functionality (due to the ActiveX control used for the Contributor application). This functionality was removed from Site Studio as a result of the browser-independent, JavaScript-based Contributor application that is used in Site Studio 10gR3 (10.1.3.3.3) and higher. See the technical reference guide for Site Studio 10gR3 for more information.

B.4.5 Assigning a Web Site Section to Your Folders

Site Studio no longer uses the Content Server folder features (Folders component) to organize and manage your site hierarchy. If a Web site created in a Site Studio release before 7.5 is upgraded, content that resides in a folder has a new metadata value ("Web Site Section") assigned to it so that it is recognized as part of the upgraded site.

Any new content added to the folder, after the upgrade, will not receive this metadata value. As such, if you want to continue using folders to add content to your site, you must assign a "Web Site Section" value to each folder.

To assign a "Web Site Section" value, perform these tasks:

1. Log onto the content server as a user with write (RW) access to the folder you want to update.
2. Open the **Browse Content** tray or menu, and then open the **Web Sites** tree.
3. Select the Web site to want to update.
4. Click **Folder Information** for the folder you want to change.
5. Select the **Update** action.
6. Click **Browse** next to the Web Site Section field.
7. Choose the corresponding Web site section.
8. Click **OK**.
9. Click **Update**.
10. Repeat these steps for each folder that you want to map to a Web site section in Site Studio.

B.4.6 Updating JSP Code

If you have created JSP code based on `SiteStudio.SSNavigationBean` and `SiteStudio.SSNavigationNode` objects, references to these objects must be changed so that "sitestudio" is all lowercase, as follows:

- `sitestudio.SSNavigationBean`
- `sitestudio.SSNavigationNode`

Index

A

access to consumption server, 5-5
Active Server Pages (ASP), 2-4, 2-8, 2-9
add-ons
 Check Out and Open, 1-2
 Content Tracker, 1-2
 Dynamic Converter, 1-2
Apache web server, 2-3, 2-5
applets in Contributor, 5-6
ASP, 2-4, 2-8, 2-9
automatic upgrade of pre-10gR4 projects, B-1

B

<base> tag, B-7
blocking access to consumption server, 5-5

C

certificate validation applet, 5-6
Check Out and Open, 1-2
clustered environments
 configuring ASP, 2-9
ClusterNodeAddress, 2-10
component
 installing the --, 2-1, 2-2
 SSUrlMapPlugin, 2-3, 2-6
component for Site Studio
 installation, 2-1, 2-2
 post-installation considerations, 2-3
 pre-installation considerations, 2-1
 uninstalling, 2-1, A-1
components
 Check Out and Open, 1-2
 DBSearchContainsOpSupport, 1-2
 IdcCommandUX, 2-9
 Site Studio, 2-1, 2-2
 SiteStudio, 1-2
configuration
 Apache, 2-3, 2-5
 Sun ONE, 2-3, 2-6
 zone fields, 2-3, 2-7
configuration variables
 DisableSiteStudioContribution, 5-5
 SSDefaultEditor, 5-5

 SSEditorDebugLevel, 5-8
 SSEnableExtranetLookCompatibility, 5-7
 SSEExtraCertificateClasses, 5-6
 SSEExtraCertificateJars, 5-7
 SSEExtraCertificateLabels, 5-6
 SSHttLayerManager, 5-7
configuring SSO, 5-7
consumption server, blocking access to --, 5-5
Content Server
 ASP, 2-4, 2-8, 2-9
 folders functionality, 1-3, B-1, B-2, B-10
 full site upgrade, B-4
 indexing, 1-2
 JSP, 2-4, 2-7, 2-8
 minimal site upgrade, B-5
 patches, 1-2
 requirements, 1-2
 search index, 1-2, 2-10, B-6
 Site Studio Certificate Validation page, 5-6
 upgrading for legacy projects, B-2
 upgrading sites, B-3
Content Tracker, 1-2
contribution graphic, 4-2
contribution mode, 4-1
 key combination, 5-4
Contributor
 access to consumption server, 5-5
 certificate validation, 5-6
 debugging, 5-8
 default editor, 5-5
 introduction, 4-1
 key combination, 5-4
 requirements, 1-3, 4-1
 starting --, 4-1
CS10gR3CoreUpdate patch, 1-2
CS10gR3NativeUpdate, 1-2
CS10gR3NativeUpdate patch, 1-2
CS752Update patch, 1-2
custom elements, 5-5, B-10
custom fragments, B-7

D

DBSearchContainsOpSupport component, 1-2
debug level in Contributor, 5-8
debugging, 5-8

default Contributor editor, 5-5
default project document information, 2-3, 2-4
Designer
 installation, 3-2
 installing --, 3-1
 introduction, 3-1
 requirements, 1-3, 3-1
 starting --, 3-2
 uninstalling --, A-1
DisableSiteStudioContribution variable, 5-5
documentation, 1-4
Dynamic Converter, 1-2

E

editor, default for Contributor, 5-5
elements, updating custom --, 5-5, B-10
enabling ASP on Content Server, 2-4, 2-8, 2-9
enabling debugging in Contributor, 5-8
enabling JSP on Content Server, 2-4, 2-7, 2-8
Ephox, 2-2, 2-4, 5-5
 and upgrading the Site Studio component, 2-2,
 2-4
 certificate validation, 5-6
 debugging, 5-8
Exclude From Lists metadata field, 5-3

F

FCKeditor, 5-5
folders on content server, 1-3, B-1, B-2, B-10
fragments, updating custom --, B-7
full site upgrade, B-4
full-text indexing, 2-3, 2-7

G

GET_SEARCH_RESULTS service, B-7, B-9

I

IdcClientLoginForm tag, 5-7
IdcCommandUX component, 2-9
IIS, 2-6
indexing, 1-2
installation
 Apache configuration, 2-3
 Apache web server, 2-3, 2-5
 ApacheSun ONE configuration, 2-5
 component, 2-1, 2-2
 default project document information, 2-3, 2-4
 Designer, 3-1
 Designer application, 3-2
 enabling ASP on Content Server, 2-4, 2-8, 2-9
 enabling JSP on Content Server, 2-4, 2-7, 2-8
 process, 1-3
 Sun ONE configuration, 2-3, 2-6
 Sun ONE web server, 2-3, 2-6
 zone fields, 2-3, 2-7

J

Java Virtual Machine (JVM), 5-6
JavaServer Pages (JSP), 2-4, 2-7, 2-8, B-11
JPS enabled groups, 2-8
JSP, 2-4, 2-7, 2-8, B-11
JVM, 5-6

K

key combination for contribution mode, 4-1, 5-4

L

launching Contributor, 4-1
launching Designer, 3-2
legacy projects, 5-8, B-1
links
 <base> tag, B-7
 SS_GET_PAGE, B-7
logging level for debugging, 5-8

M

Manager
 requirements, 1-3
metadata fields, 5-1
 Exclude From Lists, 5-3
 Region Definition, 5-4
 Web Site Object Type, 5-2
 Web Site Section, 5-2, B-10
 Web Sites, 5-1
metadata fields for Site Studio, 2-3, 2-7
metadata for project files, 2-3, 2-4
minimal site upgrade, B-5
multiple content servers, upgrading sites on --, B-3

N

NameTrans (Sun ONE web server), 2-3, 2-6
navigation on sites, B-6

P

patches
 CS10gR3CoreUpdate, 1-2
 CS10gR3NativeUpdate, 1-2
 CS752Update, 1-2
path-based URLs, 2-3, 2-5, B-7
post-installation tasks and considerations, 2-3
pre-installation tasks and considerations, 2-1
project metadata, 2-3, 2-4
projects
 upgrading pre-7.5 projects, B-1
 using pre-10gR4 projects, 5-8
proxied server, 1-2

R

rebuilding search index, B-6
rebuilding the search index, 1-2, 2-10

- redeploying JSP support, 2-8
- Region Definition metadata field, 5-4
- requirements
 - Content Server, 1-2
 - Contributor, 1-3, 4-1
 - Designer, 1-3, 3-1
 - Manager, 1-3

S

- search and indexing, 1-2
- search index, 1-2, 2-10, B-6
- search results, B-7, B-9
- services for search results, B-7, B-9
- shortcut for contribution mode, 5-4
- signed applets, 5-6
- single content server, upgrading sites on --, B-3
- single sign-on (SSO), 5-7
- site navigation, B-6
- site sections and folders, B-10
- Site Studio
 - documentation, 1-4
 - introduction, 1-1
 - metadata fields, 2-3, 2-7, 5-1
 - pre-10gR4 projects, 5-8, B-1
 - upgrades, 2-4
- Site Studio Certificate Validation page, 5-6
- Site Studio component
 - installation, 2-1, 2-2
 - post-installation considerations, 2-3
 - pre-installation considerations, 2-1
 - uninstalling, 2-1, A-1
- Site Studio metadata fields
 - Exclude From Lists, 5-3
 - Region Definition, 5-4
 - Web Site Object Type, 5-2
 - Web Site Section, 5-2, B-10
 - Web Sites, 5-1
- sites, see 'Web sites'
- SiteStudio component, 1-2
- SS_GET_PAGE, B-7
- SS_GET_SEARCH_RESULTS service, B-7, B-9
- SSDefaultEditor variable, 5-5
- SSEditorDebugLevel variable, 5-8
- SSEnableExtranetLookCompatibility variable, 5-7
- SSExtraCertificateClasses variable, 5-6
- SSExtraCertificateJars variable, 5-7
- SSExtraCertificateLabels variable, 5-6
- SSHttpLayerManager variable, 5-7
- SSNavigationBean object, B-11
- SSNavigationNode object, B-11
- SSO, 5-7
- SSUrlMapPlugin
 - component, 2-3, 2-6
- starting Contributor, 4-1
- starting Designer, 3-2
- Sun ONE web server, 2-3, 2-6
- system requirements
 - Contributor, 1-3, 4-1
 - Designer, 1-3, 3-1

- Manager, 1-3

U

- uninstalling Designer, A-1
- uninstalling Site Studio component, 2-1, A-1
- updating custom elements, 5-5, B-10
- upgrades
 - Web sites, 2-4
- upgrading legacy sites, B-2
- upgrading pre-7.5 projects
 - automated upgrade, B-1
 - <base>tag, B-7
 - custom elements, B-10
 - custom fragments, B-7
 - folders, B-10
 - full upgrade, B-4
 - GET_SEARCH_RESULTS service, B-7, B-9
 - JSP code, B-11
 - minimal upgrade, B-5
 - multiple content server, B-3
 - search index, B-6
 - single content server, B-3
 - site navigation, B-6
 - site sections, B-10
 - SS_GET_PAGE, B-7
 - upgrading your content servers, B-2
- upgrading pre-7.5 sites, B-2
- URLs
 - path-based, 2-3, 2-5, B-7

V

- validation of Ephox certificates, 5-6
- variables, see 'configuration variables'
- virtual key codes, 5-4

W

- wcm.console.htm, 5-8
- wcm.toggle.js, 5-4
- web servers
 - Apache, 2-3, 2-5
 - Sun ONE, 2-3, 2-6
- Web Site Object Type metadata field, 5-2
- Web Site Section metadata field, 5-2, B-10
- Web sites, B-2
- Web Sites metadata field, 5-1
- window.external functionality, 5-5, B-10

X

- xDontShowInListsforWebsites metadata field, 5-3
- xRegionDefinition metadata field, 5-4
- xWebsiteObjectType metadata field, 5-2
- xWebsites metadata field, 5-1
- xWebsiteSection metadata field, 5-2, B-10

Z

- zone fields, 2-3, 2-7

