

Oracle® Fusion Middleware

Using Oracle Site Studio Designer

12c (12.2.1)

E52337-01

October 2015

Oracle Site Studio Designer provides information to individuals who are responsible for the design of web sites managed by Oracle Site Studio and Oracle Content Server.

Oracle Fusion Middleware Using Oracle Site Studio Designer, 12c (12.2.1)

E52337-01

Copyright © 1996, 2015, Oracle and/or its affiliates. All rights reserved.

Primary Author: Jean Rao

Contributor: Brian Cheyne, Will Harris, Ron van de Crommert, Martin Wykes, Sean Cearley

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware 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 that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	xvii
What's New In This Guide	xix
1 Introduction	
1.1 About Oracle Site Studio	1-1
1.2 Designer, Manager, and Contributor	1-2
1.3 Oracle Content Server and Oracle Site Studio	1-2
1.4 Reusable Assets and Content	1-3
1.5 Web Sites in HCSP, JSP, and ASP	1-3
1.6 New Features and Enhancements in Oracle Site Studio 11g	1-3
1.6.1 Compared to Oracle Site Studio 10gR4 (10.1.4.5.0 - Build 9.0.0.546)	1-4
1.6.2 Compared to Oracle Site Studio 10gR4 (10.1.4.5.0 - Build 9.0.0.506)	1-4
1.6.3 Compared to Oracle Site Studio 10gR4 (10.1.4.5.0 - Build 9.0.0.470)	1-5
1.6.4 Compared to Oracle Site Studio 10gR4 (10.1.4.0.0 - Builds 9.0.0.454 and 9.0.0.410) ..	1-5
1.6.5 Compared to Oracle Site Studio 10gR4 (10.1.4.0.0 - Build 9.0.0.354)	1-6
1.6.6 Compared to Oracle Site Studio 10gR3	1-6
1.7 Accessibility Features	1-9
1.7.1 Context Menus	1-9
1.7.2 Standard Windows Keyboard Shortcuts	1-9
1.7.3 Oracle Site Studio Designer-Specific Keyboard Shortcuts	1-10
1.7.4 Fragments and Accessibility	1-10
1.8 System Requirements	1-11
2 Setting Up Oracle Site Studio Designer on Client Computers	
2.1 About Designer	2-1
2.2 System Requirements	2-1
2.3 Installing Designer	2-1
2.4 Uninstalling a Previous Version of Oracle Site Studio Designer	2-2
3 Understanding Oracle Site Studio Web Sites	
3.1 Managed Web Sites	3-1
3.2 Separation of Site Presentation and Content	3-2
3.2.1 Site Presentation Files	3-2
3.2.2 Site Content Files	3-3

3.2.3	Site Control and Configuration Files	3-3
3.3	Site Asset Storage	3-4
3.4	Site Roles	3-5
3.5	Presentation Model.....	3-6
3.6	Contribution Model.....	3-7
3.7	Site Object Hierarchy.....	3-8
3.8	Reusing Site Assets	3-12
3.9	Elements and Element Definitions	3-13
3.10	Region Templates and Region Definitions.....	3-15
3.11	Placeholders and Placeholder Definitions.....	3-17
3.12	Page Templates.....	3-20
3.13	Subtemplates.....	3-22
3.14	Index Pages and Flex Pages.....	3-23
3.15	Named Pages	3-25
3.16	Contributor Data Files and Native Documents.....	3-25
3.17	Fragments.....	3-27
3.18	Cascading Style Sheets	3-27
3.19	Project Files	3-28

4 Efficient Web Site Planning

4.1	Planning the Web Site	4-1
4.1.1	Why Is Planning Important?	4-2
4.1.2	What Parts of the Site Will Be Reused?	4-2
4.2	Naming Site Assets	4-3
4.3	Planning the Contribution Model	4-5
4.3.1	How Much of the Site Will be Provided by the User?	4-5
4.3.2	How Much Control Will the User Have?.....	4-5
4.3.3	Will There Be Different Security Access for Each User?	4-5
4.3.4	Will Contributors Submit Native Documents?	4-6
4.3.5	How Will the Contribution Process Be Coordinated?.....	4-6
4.4	Order of Site Asset Creation.....	4-7
4.5	Creating Element Definitions.....	4-7
4.6	Creating Region Definitions	4-8
4.7	Creating Region Templates	4-8
4.8	Creating Subtemplates	4-9
4.8.1	Is a Subtemplate Necessary for the Site?	4-9
4.9	Creating Placeholder Definitions.....	4-9
4.9.1	How Will The Placeholder Function On The Page?	4-9
4.9.2	What Will The Placeholder Definition Control?	4-10
4.10	Creating Page Templates	4-11
4.10.1	How Should Fragments Be Used?.....	4-11
4.10.2	Will Primary and Secondary Pages Require Different Templates?.....	4-11
4.11	Planning the Site Hierarchy.....	4-12
4.11.1	How Deep Should the Hierarchy Be?	4-13
4.11.2	How Will The User Navigate the Site?.....	4-13
4.11.3	How Should the Sections Be Named?	4-13
4.11.4	How Reusable Should The Page Templates Be?	4-13

4.11.5	Should There Be Both Primary and Secondary Pages?	4-14
4.11.6	How Will Content Be Reused?.....	4-14
4.11.7	Will A Manager Be Necessary?.....	4-15

5 Getting Started With Designer

5.1	Installing Designer	5-2
5.2	Starting Designer.....	5-2
5.3	Main Designer Window	5-3
5.4	Site Hierarchy	5-4
5.5	Properties Pane.....	5-5
5.6	Workspace.....	5-7
5.6.1	Dialogs vs. Tabs	5-7
5.6.2	Views	5-9
5.6.3	Source View	5-10
5.6.4	Design View.....	5-11
5.6.5	Preview.....	5-12
5.6.6	Form View	5-13
5.6.7	Right-Click Menus.....	5-13
5.7	Site Assets Pane	5-14
5.8	Toolbox	5-16
5.9	Menu Bar	5-17
5.9.1	File Menu	5-17
5.9.1.1	Site Submenu.....	5-17
5.9.1.2	Fragments Submenu	5-19
5.9.1.3	Definitions Submenu	5-19
5.9.2	Edit Menu	5-20
5.9.3	Formatting Menu	5-20
5.9.4	View Menu.....	5-21
5.9.5	Tools Menu	5-21
5.9.6	Window Menu	5-22
5.9.7	Help Menu	5-22
5.10	Toolbars	5-22
5.10.1	Standard Toolbar	5-23
5.10.2	Site Toolbar	5-23
5.10.3	Formatting Toolbar.....	5-24
5.10.4	HTML Toolbar	5-25
5.10.5	Table Toolbar.....	5-26
5.10.6	Contribution Toolbar	5-27
5.10.7	Placeholder Toolbar.....	5-27
5.11	Log File	5-28
5.12	Site Connection Manager.....	5-28
5.13	Link Wizard	5-30
5.14	Switch Content Wizard	5-34
5.15	Fragment Editor	5-36
5.16	Contributor Data Files and Native Documents.....	5-37
5.17	Oracle Site Studio Manager	5-39
5.18	Oracle Site Studio Contributor.....	5-40

5.18.1	Launching Contributor	5-40
5.18.2	Contributor Editor	5-41
5.18.3	Contributor and Workflows.....	5-43
5.19	Oracle Site Studio Administration Page in Oracle Content Server	5-43
5.20	Oracle Site Studio Publisher	5-45
5.20.1	Oracle Site Studio Publisher in Oracle Content Server	5-45
5.20.2	Publish Now	5-46

6 Customizing Designer

6.1	Customizing the Panes in Designer	6-1
6.1.1	Showing or Hiding Panes	6-2
6.1.2	Resizing a Pane	6-2
6.1.3	Moving or Docking a Pane to a Different Border	6-2
6.2	Using the Customize Dialog.....	6-3
6.3	Creating a New Menu	6-4
6.4	Customizing Toolbars	6-4
6.4.1	Showing and Hiding Toolbars.....	6-5
6.4.2	Creating a New Toolbar.....	6-5
6.4.3	Resetting a Toolbar	6-5
6.5	Adding Shortcuts to the Tools Menu.....	6-6
6.6	Assigning Keyboard Shortcuts	6-7
6.7	Customizing Menus	6-8
6.7.1	Adding a Command to a Popup Menu.....	6-8
6.7.2	Deleting a Command From a Popup Menu.....	6-8
6.7.3	Controlling the Display of Menus.....	6-8
6.8	Setting Display Options	6-9
6.9	Formatting the Code in Source View	6-10
6.10	Showing or Hiding Warning Messages.....	6-10
6.11	Setting Miscellaneous Options.....	6-10
6.12	Changing Log File Settings.....	6-11

7 Creating Your Web Site

7.1	Site Connections	7-1
7.1.1	Creating a New Site Connection.....	7-2
7.1.2	Editing a Site Connection	7-3
7.1.3	Deleting a Site Connection	7-4
7.2	Site Addressing	7-5
7.2.1	Mapping Domain Names to Your Site	7-5
7.2.1.1	Mapping a Domain Name Using Designer	7-6
7.2.1.2	Mapping a Domain Name Using the Manage Web Site Addresses Page.....	7-6
7.2.2	Changing the Path Used in a Site Address	7-7
7.3	Project Files	7-8
7.3.1	Specifying the Metadata That is Assigned to a Project File.....	7-8
7.3.2	Viewing the Project Status in Designer	7-9
7.3.3	Committing Changes to a Project File	7-9
7.3.4	Viewing the Content Information Page for the Project File	7-9
7.4	Starting and Stopping a Web Site	7-9

7.4.1	Stopping the Web Site	7-10
7.4.2	Configuring a Temporary Page for the Stopped Site	7-10

8 Working With Templates

8.1	About Templates	8-1
8.1.1	About Page Templates	8-2
8.1.2	About Region Templates	8-2
8.1.3	About Subtemplates	8-3
8.1.4	About Cascading Style Sheets.....	8-3
8.2	Managing Templates as Site Assets	8-3
8.2.1	Creating a New Template.....	8-4
8.2.2	Creating a Page Template From the Site Hierarchy	8-4
8.2.3	Copying a Template	8-5
8.2.4	Viewing the Content Information of a Template	8-5
8.2.5	Viewing the Content Information of a Page Template from the Site Hierarchy	8-6
8.2.6	Adding a Template to a Site	8-6
8.2.7	Removing a Template from a Site	8-6
8.3	Understanding the Contribution Regions.....	8-7
8.4	Inserting Placeholders	8-7
8.5	Inserting Objects.....	8-8
8.6	Inserting Fragments.....	8-8
8.7	Working With Text	8-8
8.8	Arranging Objects on the Template	8-9
8.9	Applying CSS Classes	8-9
8.10	Working With HTML Tables.....	8-10
8.11	Viewing Templates in Designer.....	8-11
8.11.1	Design View.....	8-11
8.11.2	Source View	8-12
8.11.3	Preview.....	8-13
8.12	Region Templates and Region Definitions.....	8-14
8.13	Working With Native Documents and Conversion Definitions.....	8-14

9 Working With Definitions

9.1	About Definitions.....	9-1
9.1.1	About Placeholder Definitions	9-1
9.1.2	About Region Definitions	9-2
9.1.3	About Element Definitions.....	9-2
9.2	Creating a New Definition.....	9-2
9.3	Editing Definitions.....	9-3
9.3.1	Editing a Placeholder Definition	9-3
9.3.2	Editing a Region Definition.....	9-4
9.3.3	Editing an Element Definition	9-5
9.4	Copying a Definition	9-5
9.5	Viewing the Content Information of a Definition.....	9-6
9.6	Adding a Definition to a Site	9-7
9.7	Removing a Definition from a Site	9-7

9.8	Working With Placeholder Definitions	9-7
9.8.1	Specifying a Default Placeholder Definition.....	9-8
9.8.2	Creating a Global Placeholder Definition Mapping	9-9
9.8.3	Creating a Section Property Placeholder Definition Mapping	9-9
9.8.4	Specifying a Placeholder Definition Directly in the Placeholder Tag.....	9-10
9.8.5	Associating a Placeholder with a Global Placeholder Definition	9-11
9.8.6	Adding a Region Definition to a Placeholder Definition.....	9-11
9.8.7	Associating a Region Template with a Placeholder Definition	9-12
9.8.8	Associating a Subtemplate with a Placeholder Definition	9-13
9.9	Working With Region Definitions.....	9-13
9.10	Working With Element Definitions.....	9-15
9.11	Viewing Definitions in Designer	9-15
9.11.1	Form View	9-15
9.11.2	Source View	9-16
9.12	Definition Bundles	9-17
9.12.1	About Definition Bundles.....	9-17
9.12.2	Downloading Bundles	9-17
9.12.3	Uploading Bundles.....	9-18

10 Working With Content Files

10.1	About Content Files.....	10-1
10.1.1	Contributor Data Files.....	10-1
10.1.2	Native Documents	10-2
10.1.3	Conversion Definitions	10-2
10.1.4	Images.....	10-2
10.2	Creating a New Content File.....	10-2
10.3	Copying a Content File	10-3
10.4	Viewing the Content Information of a Content File	10-3
10.5	Adding a Content File to a Site	10-4
10.6	Removing a Content File from a Site	10-4
10.7	Editing a Contributor Data File	10-4
10.8	Working With Native Documents.....	10-5
10.8.1	Requirements for Native Documents	10-5
10.8.2	Editing a Native Document.....	10-6
10.8.3	Using a Conversion Definition With Native Documents	10-6
10.8.4	Using Dynamic Converter Template Selection Rules	10-7
10.8.5	Sample Fragments for Native Documents	10-8
10.8.6	Using Direct Delivery.....	10-9
10.9	Working With Images	10-9

11 Working With Scripts and Forms

11.1	About Scripts and Forms	11-1
11.1.1	About Scripts	11-1
11.1.2	About Custom Configuration Scripts	11-2
11.1.3	About Custom Element Forms	11-2
11.1.4	About Validation Scripts	11-2
11.2	Managing Assets	11-2

11.2.1	Creating a New Script	11-2
11.2.2	Copying a Script.....	11-3
11.2.3	Editing a Script.....	11-4
11.2.4	Viewing the Content Information of a Script	11-4
11.2.5	Adding a Script to a Site	11-4
11.2.6	Removing a Script from a Site	11-5
11.3	Custom Element Forms.....	11-5
11.3.1	Implementing a Custom Element.....	11-5
11.3.2	Backward Compatibility and Upgrading.....	11-12
11.3.3	Sample Custom Elements.....	11-13
11.4	Custom Configuration Scripts	11-14
11.5	Default Validation Options	11-15

12 Building a Site

12.1	Working With the Site Hierarchy	12-1
12.1.1	Adding a Section to the Site Hierarchy	12-2
12.1.2	Renaming a Section in the Site Hierarchy	12-3
12.1.3	Rearranging Sections in the Site Hierarchy	12-3
12.1.4	Removing a Section From the Site Hierarchy	12-4
12.1.5	Assigning a Home Page.....	12-4
12.1.6	Assigning a Primary Page	12-5
12.1.7	Assigning a Secondary Page	12-6
12.1.8	Showing Sections in the Site Navigation.....	12-7
12.1.9	Designating a Section as an Error Handler	12-9
12.1.10	Changing the Security of the Site or a Section.....	12-9
12.1.11	Viewing and Editing Web Site and Section Properties	12-10
12.1.12	Defining Custom Properties for a Section.....	12-12
12.1.13	Expanding and Collapsing the Site Hierarchy	12-13
12.1.14	Refreshing Your Site Hierarchy in Designer	12-14
12.2	Working With Fragments	12-14
12.3	Assigning Content With The Switch Content Wizard	12-14
12.4	Working With Links	12-16
12.4.1	Creating a Link to Another Section of the Web Site	12-16
12.4.2	Creating a Link to an Existing File on the Content Server	12-17
12.4.3	Creating a Link to a URL	12-19
12.4.4	Creating a Link to a New Content File	12-20
12.4.5	Creating a Link to an Existing Local File.....	12-21
12.4.6	Editing a Hyperlink.....	12-23
12.4.7	Removing a Link.....	12-24
12.4.8	Choosing a Default Link Format	12-24
12.4.9	Hyperlink Formats	12-25
12.5	Comparing Changes on a Web Page.....	12-26
12.5.1	Color-Coded Differences when Comparing Web Pages.....	12-27
12.5.2	Using the Compare Changes Feature	12-27

13 Working With Fragments

13.1	About Fragments	13-1
13.2	Adding and Editing Fragments	13-2
13.3	Fragment Handling in Templates.....	13-2
13.4	Fragments in the Toolbox	13-2
13.5	Filtering Fragments in the Toolbox	13-4
13.6	Adding Fragments to a Template.....	13-4
13.7	Previewing Fragments	13-5
13.7.1	Previewing a Fragment.....	13-5
13.7.2	Choosing Different Parameters for a Fragment	13-6
13.7.3	Choosing Different Parameters When Adding a Fragment	13-6
13.7.4	Changing the Parameters of a Fragment.....	13-6
13.8	Editing Fragments.....	13-7
13.8.1	Editing a Copy of a Fragment	13-7
13.8.2	Editing a Fragment	13-8
13.9	Creating a New Fragment	13-9
13.10	Deleting a Fragment	13-10
13.11	Deleting a Fragment in the Toolbox.....	13-10
13.12	Using the Fragment Editor	13-10
13.12.1	Opening the Fragment Editor	13-11
13.12.2	Specifying Fragment Properties	13-12
13.12.3	Changing the Icon Associated with a Fragment	13-13
13.12.4	Adding, Editing, and Deleting Fragment Snippets	13-14
13.12.5	Adding, Editing, and Deleting Fragment Assets	13-15
13.12.5.1	Adding an Asset	13-16
13.12.5.2	Editing an Asset.....	13-16
13.12.5.3	Deleting an Asset.....	13-17
13.12.6	Adding, Editing, and Deleting Fragment Parameters.....	13-17
13.12.6.1	Editing the Parameter in a Fragment.....	13-18
13.12.6.2	Adding a New Parameter to a Fragment	13-19
13.12.6.3	Inserting Parameters and Parameter Declarations in a Snippet.....	13-19
13.12.7	Adding, Editing, and Deleting Fragment Elements	13-20
13.12.7.1	Adding an Element to a Fragment.....	13-20
13.12.7.2	Editing an Element in a Fragment	13-21
13.12.7.3	Deleting an Element in a Fragment	13-21
13.13	Fragment Libraries.....	13-22
13.13.1	About Fragment Libraries	13-22
13.13.2	Saving a Fragment in a Fragment Library	13-23
13.13.3	Making a Fragment Library Read-Only	13-24
13.13.4	Uploading and Downloading Fragment Libraries	13-24
13.13.4.1	Downloading a Fragment Library	13-25
13.13.4.2	Uploading a Fragment Library.....	13-25

14 Working With Site Reports

14.1	About Site Reports	14-1
14.2	Web Site Objects Report.....	14-1
14.3	Site Usage Report	14-2

14.4	Viewing a Web Site Objects Report.....	14-3
14.4.1	Viewing the Report in Designer	14-3
14.4.2	Viewing the Report on the Oracle Site Studio Administration Page.....	14-4
14.5	Viewing a Web Site Usage Report.....	14-4
14.5.1	Viewing a Report for a Page Template in Designer	14-4
14.5.2	Viewing a Report for a Data File or Native Document in Designer.....	14-4
14.5.3	Viewing a Report for a Data File or Native Document From the Web Page	14-5
14.5.4	Viewing a Report for All Other Files	14-5

15 Using Workflows

15.1	About Workflows.....	15-1
15.2	Workflow Types.....	15-1
15.3	Workflow Experience for Contributors	15-2
15.4	Setting Up a Workflow.....	15-2
15.5	Workflow Examples	15-3
15.5.1	Contributor Data Files in Workflows	15-3
15.5.2	Native Documents in Workflows.....	15-3
15.5.3	Images in Workflows	15-4

16 Setting Up Manager

16.1	About Oracle Site Studio Manager.....	16-1
16.2	Choosing Where and When to Display Manager	16-2
16.3	When Manager Opens.....	16-2
16.4	Adding Manager to Your Site	16-2
16.4.1	Adding Manager With Default Settings.....	16-2
16.4.2	Adding Manager With Custom Settings.....	16-3
16.5	Creating a Manager Configuration Settings File.....	16-4
16.6	Copying a Manager Configuration Settings File.....	16-4
16.7	Customizing Manager Configuration Settings.....	16-5
16.8	Viewing the Content Information of a Manager Configuration Settings File.....	16-7
16.9	Adding a Manager Configuration Settings File to a Site	16-7
16.10	Removing a Manager Configuration Settings File from a Site	16-7
16.11	Description of the Manager Configuration Settings File	16-8

A User Interface

A.1	Customize Dialog.....	A-3
A.1.1	Customize Dialog: Commands Tab	A-3
A.1.2	Customize Dialog: Toolbars Tab	A-4
A.1.3	Customize Dialog: Tools Tab	A-5
A.1.4	Customize Dialog: Keyboard Tab	A-5
A.1.5	Customize Dialog: Menu Tab	A-6
A.1.6	Customize Dialog: Options Tab	A-6
A.1.7	Customize Dialog: Source View Tab	A-7
A.1.8	Customize Dialog: Warning Dialogs Tab.....	A-8
A.1.9	Customize Dialog: Miscellaneous Tab.....	A-8
A.1.10	Customize Dialog: Log File Tab	A-8

A.2	Site Connection Manager Dialog	A-9
A.3	Site Connection Details Dialog	A-9
A.4	Create New Site Dialog	A-10
A.5	Site Addresses Dialog.....	A-10
A.6	Choose Default Link Format Dialog	A-11
A.7	Define Environment Properties Dialog	A-12
A.8	New Section Dialog	A-12
A.9	Custom Section Properties Dialog.....	A-13
A.10	Define Custom Section Property Dialog.....	A-13
A.11	Select (Home) Page Template Dialog.....	A-14
A.12	Select (Primary) Page Template Dialog	A-15
A.13	Select (Secondary) Page Template Dialog	A-16
A.14	Site Asset Categories Dialog.....	A-17
A.15	Enable Metadata Modification Dialog	A-18
A.16	Select Site Dialog	A-19
A.17	Select Section Dialog.....	A-19
A.18	Link Wizard	A-19
A.18.1	Link Wizard - Choose Type of Link.....	A-20
A.18.2	Link Wizard - Choose a Section.....	A-20
A.18.3	Link Wizard - Choose URL Format	A-20
A.18.4	Link Wizard - Choose Content File.....	A-21
A.18.5	Link Wizard - Choose Target Section	A-22
A.18.6	Link Wizard - Check-In Content	A-22
A.18.7	Link Wizard - Confirmation.....	A-23
A.18.8	Link Wizard - Link Target Options.....	A-23
A.19	Contribution Region Dialog	A-23
A.20	Region Content Options Dialog.....	A-24
A.21	Choose Document Types Dialog	A-25
A.22	Edit Query Text Dialog	A-25
A.23	Conversion Definition Dialog	A-26
A.24	Native Document Conversion Settings Dialog (Legacy)	A-26
A.25	Native Document Conversion Settings Dialog.....	A-27
A.26	WYSIWYG Element Dialog	A-28
A.27	WYSIWYG Element Definition Dialog	A-31
A.28	Plain Text Element Dialog	A-34
A.29	Plain Text Element Definition Dialog	A-35
A.30	Image Element Dialog.....	A-36
A.31	Image Element Definition Dialog	A-37
A.32	Custom Element Dialog	A-39
A.33	Custom Element Definition Dialog	A-39
A.34	Custom Element Settings Dialog.....	A-40
A.35	Static List Element Dialog.....	A-40
A.36	Static List Element Definition Dialog.....	A-42
A.37	Elements Dialog	A-42
A.38	Static List Element Validation Dialog	A-43
A.39	Dynamic List Element Dialog	A-43
A.40	Dynamic List Element Definition Dialog	A-45

A.41	Dynamic List Settings Dialog.....	A-46
A.42	Region Definition Dialog	A-46
A.43	Element Dialog	A-47
A.44	Link Settings Dialog	A-47
A.45	Select CSS Classes Dialog	A-48
A.46	Select a CSS StyleSheet Dialog	A-49
A.47	Choose Fragment Asset Dialog.....	A-49
A.48	WYSIWYG Element Validation Dialog.....	A-49
A.49	Plain Text Element Validation Dialog.....	A-50
A.50	Select HTML Tags Dialog	A-51
A.51	Image Element Validation Dialog	A-51
A.52	Advanced Element Validation Dialog	A-52
A.53	New Item Settings Dialog	A-52
A.54	Assign Content Dialog	A-53
A.55	Choose Region Content Dialog.....	A-54
A.56	Generate Unique Region Content Dialog.....	A-54
A.57	Fragment Editor Dialog	A-55
A.58	Properties for Fragment Dialog	A-56
A.59	Fragment Assets Dialog	A-57
A.60	Fragment Elements Dialog	A-57
A.61	Fragment Parameters Dialog.....	A-58
A.62	Edit Query Text Dialog	A-60
A.63	New Option Dialog.....	A-60
A.64	Snippet Properties Dialog.....	A-61
A.65	Fragment Parameter Values Dialog	A-61
A.66	Fragment Preview Dialog.....	A-62
A.67	Save Fragment Dialog	A-63
A.68	Manage Site Page	A-63
A.69	Manager Configuration Settings Dialog.....	A-64
A.70	Placeholder Section Properties Dialog.....	A-65
A.71	Edit Placeholder Section Properties Dialog	A-65
A.72	Placeholder Definition Dialog.....	A-66
A.73	Placeholder Definition Mappings Dialog.....	A-67
A.74	Define Placeholder Definition Mapping Dialog.....	A-68
A.75	Define Placeholder Dialog.....	A-68
A.76	Switch Content Wizard	A-68
A.76.1	Switch Content Wizard: Choose Region Content Screen	A-68
A.76.2	Switch Content Wizard: Choose Content File Screen	A-69
A.76.3	Switch Content Wizard: Check-In Content Screen.....	A-69
A.76.4	Switch Content Wizard: Confirmation Screen	A-70
A.77	Switch Region Template Dialog	A-70
A.78	Insert Dynamic Conversion Dialog.....	A-70
A.79	Custom Configuration Scripts Dialog.....	A-70
A.80	Contribution Element Dialog.....	A-71
A.81	Select Fragment Dialog	A-71
A.82	Oracle Site Studio Replicator.....	A-72
A.82.1	Step 1: Select Source Server	A-72

A.82.2	Step 2: Select Target Server.....	A-73
A.82.3	Step 3: Select Items to Replicate.....	A-73
A.82.4	Step 4: Perform Replication.....	A-74
A.82.5	Modifying Server Connections.....	A-74
A.82.6	Advanced Options.....	A-74
A.82.7	Override Environment Properties Dialog.....	A-75

B Replicating Your Site

B.1	About Replication.....	B-1
B.1.1	The Replication Process.....	B-2
B.1.2	What Gets Replicated.....	B-3
B.1.3	Revisions Included.....	B-3
B.1.4	Additional Export Settings.....	B-3
B.2	Configuring the Target Server for Replication.....	B-3
B.3	Setting Up an Outgoing Provider on the Source Server.....	B-4
B.4	Creating the Import and Export Archives.....	B-6
B.4.1	Setting Up the Import Archive on the Target Server.....	B-6
B.4.2	Setting Up the Export Archive on the Source Server.....	B-7
B.5	Starting the Site Replication.....	B-8
B.6	Specifying Environment Properties.....	B-9
B.7	Replication Individual Sections of Your Site.....	B-9
B.8	Using Oracle Site Studio Replicator.....	B-10
B.8.1	Step 1: Selecting a Source Server and Web Site.....	B-10
B.8.2	Step 2: Selecting a Target Server.....	B-11
B.8.3	Step 3: Selecting Items to Replicate.....	B-13
B.8.4	Step 4: Completing Oracle Site Studio Replicator.....	B-14
B.8.5	Modifying Server Connections.....	B-15
B.8.6	Overriding Environment Properties.....	B-16

C Sample Fragments

C.1	Navigation Fragments.....	C-1
C.1.1	Breadcrumb Plain.....	C-2
C.1.2	Explorer Menu Bar.....	C-3
C.1.3	Login Example.....	C-3
C.1.4	Nav Home Logo.....	C-3
C.1.5	Nav Multi Horizontal.....	C-3
C.1.6	Nav Multi Vertical.....	C-4
C.1.7	Nav Plain Horizontal.....	C-4
C.1.8	Nav Plain Vertical.....	C-4
C.1.9	Nav Tabs Top.....	C-5
C.1.10	Nav Wrap Horizontal.....	C-5
C.1.11	Search Box Plain.....	C-6
C.1.12	Search Result Paged.....	C-6
C.1.13	Search Result Plain.....	C-7
C.1.14	Site Map Plain.....	C-7
C.1.15	Site Map Tree.....	C-8
C.1.16	Vertical Menu Bar.....	C-8

C.1.17	ASP Bread Crumb Plain.....	C-8
C.1.18	ASP Bread Crumb Plain (server).....	C-8
C.1.19	ASP Explorer Menu Bar.....	C-9
C.1.20	ASP Nav Multi Horizontal.....	C-9
C.1.21	ASP Nav Multi Vertical.....	C-9
C.1.22	ASP Search Box Plain.....	C-10
C.1.23	ASP Search Box Plain (server).....	C-10
C.1.24	ASP Search Result Plain.....	C-11
C.1.25	ASP Site Map Plain.....	C-11
C.1.26	CSP Sample Navigation (client).....	C-12
C.1.27	CSP Sample Navigation (server).....	C-13
C.1.28	JSP Nav Home Logo.....	C-13
C.1.29	JSP Nav Plain Horizontal.....	C-13
C.1.30	JSP Nav Plain Vertical.....	C-14
C.1.31	JSP Search Box Plain.....	C-14
C.1.32	JSP Search Results Paged.....	C-14
C.1.33	JSP Site Map Plain.....	C-15
C.1.34	IDOC Breadcrumb Plain.....	C-16
C.1.35	IDOC Nav Home Logo.....	C-16
C.1.36	IDOC Search Box Plain.....	C-16
C.1.37	RSET Breadcrumb Plain.....	C-16
C.1.38	RSET Nav Multi Horizontal.....	C-17
C.1.39	RSET Nav Plain Horizontal.....	C-17
C.1.40	RSET Site Map Plain.....	C-18
C.1.41	Additional IDOC Fragments.....	C-18
C.1.41.1	IDOC Nav Multi Horizontal.....	C-18
C.1.41.2	IDOC Nav Plain Horizontal.....	C-18
C.1.41.3	IDOC Nav Tabs Top.....	C-19
C.1.41.4	IDOC Site Map Plain.....	C-19
C.2	Dynamic List Fragments.....	C-20
C.2.1	Dynamic List Generic.....	C-20
C.2.2	Dynamic List Paged.....	C-20
C.2.3	Dynamic List Plain.....	C-20
C.2.4	ASP Dynamic List Paged.....	C-21
C.2.5	ASP Dynamic List Simple.....	C-22
C.2.6	CSP Sample Dynamic List.....	C-22
C.2.7	JSP Dynamic List Plain.....	C-22
C.3	Static List Fragments.....	C-23
C.3.1	Static List Example.....	C-23
C.3.2	Static List Generic.....	C-23
C.3.3	ASP Static List Example.....	C-24
C.4	Other Fragments.....	C-24
C.4.1	Copyright Example.....	C-24
C.4.2	Dynamic Conversion.....	C-25
C.4.3	Dynamic Conversion IFrame.....	C-25
C.4.4	Error Handler Message.....	C-25
C.4.5	Flash Example.....	C-25

C.4.6	Logo Example.....	C-26
C.4.7	Random Conversion.....	C-26
C.4.8	ASP Error Handler Message	C-26
C.4.9	CSP Sample Page Title	C-26
C.4.10	PageHitCount.....	C-27
C.4.11	Oracle Site Studio Manager.....	C-27

D Design Mode

D.1	About Design Mode	D-1
D.2	Making Design Mode Available.....	D-1
D.3	Using Design Mode	D-2
D.3.1	Entering Design Mode	D-2
D.3.2	Working with Primary Pages.....	D-3
D.3.3	Working with Secondary Pages.....	D-3
D.4	Using the Placeholder Section Properties Dialog.....	D-4

Glossary

Preface

Using Oracle Site Studio Designer contains information to assist individuals who are responsible for the design of web sites managed by Oracle Site Studio and Oracle Content Server.

Audience

This document is intended for those people identified in the organization who are responsible for designing an organization's web site managed by Oracle Site Studio.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Site Studio documentation set:

- *Using Oracle Site Studio Contributor*
- *Using Oracle Site Studio Publisher*
- *Managing Oracle Site Studio*
- *Technical Reference for Oracle Site Studio*

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.

Convention	Meaning
<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.

What's New In This Guide

This section summarizes the new features and significant product changes for *Oracle Site Studio Designer* in the Oracle Fusion Middleware 12c (12.2.1) release.

- [New and Changed Features for Release 12c \(12.2.1\)](#)

New and Changed Features for Release 12c (12.2.1)

- The Oracle WebCenter Content Tracker Reports have been deprecated in this release.

New and Changed Features for Release 11g R1 (11.1.1)

- [Section 1.6.1, "Compared to Oracle Site Studio 10gR4 \(10.1.4.5.0 - Build 9.0.0.546\)"](#)
- [Section 1.6.2, "Compared to Oracle Site Studio 10gR4 \(10.1.4.5.0 - Build 9.0.0.506\)"](#)
- [Section 1.6.3, "Compared to Oracle Site Studio 10gR4 \(10.1.4.5.0 - Build 9.0.0.470\)"](#)
- [Section 1.6.4, "Compared to Oracle Site Studio 10gR4 \(10.1.4.0.0 - Builds 9.0.0.454 and 9.0.0.410\)"](#)
- [Section 1.6.5, "Compared to Oracle Site Studio 10gR4 \(10.1.4.0.0 - Build 9.0.0.354\)"](#)
- [Section 1.6.6, "Compared to Oracle Site Studio 10gR3"](#)

Introduction

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

This section covers the following topics:

- [Section 1.1, "About Oracle Site Studio"](#)
- [Section 1.2, "Designer, Manager, and Contributor"](#)
- [Section 1.3, "Oracle Content Server and Oracle Site Studio"](#)
- [Section 1.4, "Reusable Assets and Content"](#)
- [Section 1.5, "Web Sites in HCSP, JSP, and ASP"](#)
- [Section 1.6, "New Features and Enhancements in Oracle Site Studio 11g"](#)
- [Section 1.7, "Accessibility Features"](#)
- [Section 1.8, "System Requirements"](#)

1.1 About Oracle Site Studio

Oracle 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. Oracle Site Studio dramatically improves productivity and enables organizations to maintain accurate, timely, and current web content with consistent branding and presentation across all corporate sites. With Oracle Site Studio, you can centralize control of site architecture and presentation, while distributing content development and ongoing maintenance to business units or other teams.

Note: Oracle Site Studio 11gR1 is fully backward compatible to 10gR4. Oracle Site Studio Designer 11gR1 can also work with sites created in Oracle Site Studio releases 10gR3 and earlier. It is important to note, though, that sites created with Oracle Site Studio versions 10gR3 and earlier continue to work in "legacy" mode; that is, they can not take advantage of the architecture and features of Oracle Site Studio 11gR1.

1.2 Designer, Manager, and Contributor

Oracle Site Studio comprises three applications, which reflect the three distinct users of Oracle Site Studio: designers, managers, and contributors.

The **Oracle Site Studio Designer** application provides the development environment in which a single individual (the site designer) can create, design, and distribute the site. The designer is typically a webmaster, a web developer, a site administrator, or someone in a similar position. Typically, one designer works with multiple managers and contributors.

The **Oracle Site Studio Manager** application provides a web-based site management console that can be used by one or more site managers to maintain the structure of the site. The manager is often the head of a division or department. See *Managing Oracle Site Studio* for more information.

The **Oracle Site Studio Contributor** application enables assigned individuals within an organization (the contributors) to update the content on the web site whenever they want using an inline editing environment that can be called directly from the web site. Contributors are usually non-technical users and might be members of the marketing team or sales team. See *Using Oracle Site Studio Contributor* for more information.

Designers of the site spend most of their time in the Designer application, using the Manager and Contributor applications mostly to preview or test the site. Managers spend most of their time in Manager and may occasionally use Contributor to make minor edits. Contributors most likely work exclusively in Contributor. Of course, the same individual can design, manage, and contribute to the web site (a more likely scenario in a smaller organization).

Even though these applications are separate, they are tightly integrated. The tasks performed in Designer affect Manager and Contributor, and tasks performed in Manager affect Contributor. For example, setting up a Manager fragment or assigning content to pages in Designer, or modifying the editing menu available to the contributors, directly affect the appearance and behavior of Manager and Contributor.

1.3 Oracle Content Server and Oracle Site Studio

Oracle Content Server is the main repository for your web site. Oracle Content Server enables everyone in your organization to easily contribute content, efficiently manage the content with rich library services, and securely access content anywhere.

As part of Universal Content Management (UCM), Oracle Site Studio employs many of the built-in services that the content server has to offer, such as managed content, Idoc Script, security, and workflow. You can use the Oracle Site Studio Administration page on the content server to administer several tasks related to Oracle Site Studio.

Oracle Site Studio also works in tandem with other components, like Dynamic Converter and Check Out and Open, which enable users to seamlessly incorporate native documents (Microsoft Word, Excel, PowerPoint, and so forth) on their web site. Dynamic Converter is used to convert native documents into web pages that then appear on the web site. Check Out and Open is used to provide an in-context "check out" option directly from the web page where the document appears. Additionally, this option will launch the native application and open the file for editing.

You may also choose to enable Oracle Content Server's folders functionality or other capabilities to make it easier for contributors to submit content to the site. Experience with Oracle Content Server and its add-ons is helpful in creating and managing web sites with Oracle Site Studio. Please see the Oracle Content Server documentation for more information.

1.4 Reusable Assets and Content

Most web sites begin with the creation of a single HTML or scripting-based web page. The look and feel of the site is designed and site navigation is determined. Content, typically a mixture of text and graphics, is added last. As the site grows, this first page can be saved under a new name and essentially serve as a "template" for future web pages, all of which are connected.

This approach works well until you find yourself making a global change to the web site, which could require individual edits to every page on the site. To do this, many large web sites take a modular approach that uses dynamic includes and database-driven web pages. As a result, portions of content are assembled to display as a single web page when the web browser requests it. Thus, site navigation, advertisements, headers and footers, and information that changes frequently can be updated just once and take effect immediately. However, this requires a significant development effort and a lot of coordination.

Oracle Site Studio takes a similar approach, but it goes one step further by offering reusable layout and content. Some site assets, such as page templates, region templates, and placeholders, are used to provide background information ("look and feel") for a typical web page, while other assets, such as region definitions and element definitions, can modify and filter the layout and content, leaving much of the content of the page open for contributors to create and edit within the given page. Individual portions of the web page (both content and layout) can be made reusable, allowing those assets to be used and reused multiple times in the same way the page templates themselves are used and reused.

1.5 Web Sites in HCSP, JSP, and ASP

Before you create your web site in Oracle Site Studio, you should be aware that HCSP sites are the only site type that can take advantage of the architecture and features in Oracle Site Studio 11gR1. Legacy (that is, pre-10gR4) sites continue to work as they did with Oracle Site Studio 11gR1, but you are strongly encouraged to convert your old HCSP web sites to take advantage of the 11gR1 architecture and features. ASP and JSP sites are still available for legacy (that is, pre-10gR4) projects, but they do not have the 11gR1 functionality. They will function as they did with previous releases of Oracle Site Studio.

Please note the following:

- Idoc and JSP are grouped as "hcsp/jsp" because they can both execute in a content server environment (JSP pages use the Java ServerBean to call Idoc services). ASP, however, exists by itself because it cannot execute in such an environment.
- While working in Designer, you can hide the fragments in the Toolbox that are not written in the same language as your web site.

1.6 New Features and Enhancements in Oracle Site Studio 11g

Oracle Site Studio 11g Release 1 (11.1.1) has many new features and enhancements compared to previous releases:

- [Section 1.6.1, "Compared to Oracle Site Studio 10gR4 \(10.1.4.5.0 - Build 9.0.0.546\)"](#)
- [Section 1.6.2, "Compared to Oracle Site Studio 10gR4 \(10.1.4.5.0 - Build 9.0.0.506\)"](#)
- [Section 1.6.3, "Compared to Oracle Site Studio 10gR4 \(10.1.4.5.0 - Build 9.0.0.470\)"](#)

- [Section 1.6.4, "Compared to Oracle Site Studio 10gR4 \(10.1.4.0.0 - Builds 9.0.0.454 and 9.0.0.410\)"](#)
- [Section 1.6.5, "Compared to Oracle Site Studio 10gR4 \(10.1.4.0.0 - Build 9.0.0.354\)"](#)
- [Section 1.6.6, "Compared to Oracle Site Studio 10gR3"](#)

1.6.1 Compared to Oracle Site Studio 10gR4 (10.1.4.5.0 - Build 9.0.0.546)

Oracle Site Studio 11gR1 includes these enhancements over the 10.1.4.5.0 release of Oracle Site Studio.

Section Level Replication

Oracle Site Studio now features a method of marking certain sections of a web site as not ready to move, so they would not be included in replication. This allows a section to be marked as not ready for replication while significant changes are being made to the section. When the section is ready to be replicated the property can be reset in Designer, Manager or directly from the page in contribution mode. When this feature is enabled, adding a new section or performing a switch content on a section will automatically set this value to not ready so that the new content will not automatically be replicated to another server.

1.6.2 Compared to Oracle Site Studio 10gR4 (10.1.4.5.0 - Build 9.0.0.506)

Oracle Site Studio 11gR1 includes the enhancements listed in [Section 1.6.1, "Compared to Oracle Site Studio 10gR4 \(10.1.4.5.0 - Build 9.0.0.546\)"](#), as well as the following enhancements.

Enable Contributor Console from Administration Page

In previous versions of Oracle Site Studio, you had to manually move a file from one folder to another in order to enable the Contributor console window. Now there is a button in the Administrator pages that moves the file for you when you want to enable the console.

Definition Bundles

Oracle Site Studio now features a method for bundling a collection of related definitions and templates to replicate to other web sites. This allows you to reuse a 'set' of (for example) definitions and templates that make up a specific method of laying out information without having to manually re-create each individual definition and template. The bundle of definitions and templates can be copied in one unit to another web site to use.

New Contributor Appearance

Oracle Site Studio Contributor now features a new interface, which optimizes many of the editing and workflow tasks in Contributor.

Remove Region Content on Edit Menu

The Contribution Region menu in Oracle Site Studio Contributor now opens (when enabled in Oracle Site Studio Designer) a Remove Region Content option. This allows the contributor to quickly remove the content rather than going through the Switch Content Wizard.

Oracle Site Studio Component Reset

Oracle Site Studio now has an option to reset the component update numbering. This allows you to redeploy the original assets of an installation independent of the base content server installation, if you find it necessary.

Contribution Mode URL

The contribution mode in Oracle Site Studio Contributor can be opened when you add `/contributor/` to the URL. This is used mainly for those creating third-party interfaces for contribution mode. A configuration flag must be set for the URL method to work.

Read-Only Mode for Blocked Assets

Oracle Site Studio Designer now alerts you on opening a site asset if it is opened elsewhere for editing, and gives you the option of opening it as read-only. Previously the asset could not be opened if someone else was editing the asset.

DAM Rendition

Oracle Site Studio now works with Digital Asset Management renditions, when the component is installed on the content server.

1.6.3 Compared to Oracle Site Studio 10gR4 (10.1.4.5.0 - Build 9.0.0.470)

Oracle Site Studio 11gR1 includes the enhancements listed in [Section 1.6.2, "Compared to Oracle Site Studio 10gR4 \(10.1.4.5.0 - Build 9.0.0.506\)"](#), as well as the following enhancements.

Design Mode

Oracle Site Studio now has an additional method of working with the web site as a Designer. Design Mode is a method of granting some of the tools available in Oracle Site Studio Designer through the Contributor interface. This function is especially useful to those who are using the Oracle Site Studio extension for JDeveloper.

On-Demand Editor

Oracle Site Studio now features a method of speeding the load time of Contributor when a page has a large number of elements. When on-demand editors are enabled, a preview of the data in each element opens, and you click on the element to edit the information.

Page Name Control

Oracle Site Studio now has the ability to change the default URL page name from "index.htm".

Direct Delivery

Oracle Site Studio now allows for direct delivery of native documents. Native documents can now be easily delivered to download to a local computer rather than being published through conversion to the Web page.

1.6.4 Compared to Oracle Site Studio 10gR4 (10.1.4.0.0 - Builds 9.0.0.454 and 9.0.0.410)

Oracle Site Studio 11gR1 includes the enhancements listed in [Section 1.6.2, "Compared to Oracle Site Studio 10gR4 \(10.1.4.5.0 - Build 9.0.0.506\)"](#), [Section 1.6.3, "Compared to Oracle Site Studio 10gR4 \(10.1.4.5.0 - Build 9.0.0.470\)"](#), as well as the following enhancements.

JavaScript and CSS Compression

Oracle Site Studio features a method of compressing and concatenating the CSS and JavaScript used on the web site. This method reduces both the number and the size of the JavaScript and CSS files that must be downloaded by the browser while loading Contributor.

Workflow Approve All

Oracle Site Studio now features a method of approving all items in a person's workflow that are on one Web page with one click.

1.6.5 Compared to Oracle Site Studio 10gR4 (10.1.4.0.0 - Build 9.0.0.354)

Oracle Site Studio 11gR1 includes the enhancements listed in [Section 1.6.2, "Compared to Oracle Site Studio 10gR4 \(10.1.4.5.0 - Build 9.0.0.506\),"](#) [Section 1.6.3, "Compared to Oracle Site Studio 10gR4 \(10.1.4.5.0 - Build 9.0.0.470\),"](#) and in [Section 1.6.4, "Compared to Oracle Site Studio 10gR4 \(10.1.4.0.0 - Builds 9.0.0.454 and 9.0.0.410\),"](#) as well as the following enhancements.

On-Demand Web Sites

Oracle Site Studio administrators can now mark web sites as served "on demand." The web sites marked as on-demand web sites are not loaded into the content server's memory until a request is received to view the site or edit the site in Designer. Using on-demand web sites prevents tying up content server resources for sites that are rarely accessed, and reduces load times when starting the content server.

The Manage Web Sites page on the content server includes an On-Demand Web Sites button, which opens a page where you can mark sites as "on-demand" sites. There will be a slight delay to serve those sites listed as on-demand when accessed by a user for the first time.

Domain and Folder-Based Addressing Changes

Oracle Site Studio can now identify web sites by a combination of folders and domains. Previous versions of Oracle Site Studio required that you use either a folder-based or a domain-based address. Administrators can now specify a domain, a folder, or both to map to a web site.

1.6.6 Compared to Oracle Site Studio 10gR3

In addition to the enhancements listed compared to 10gR4 releases, Oracle Site Studio 12c (12.2.1) includes the following new features and enhancements, compared to 10gR3 releases.

Improved Reusability of Site Assets and Content

Oracle Site Studio has gone through a complete internal redesign to facilitate the reusability of site assets and content. The way that a web site is designed and built in Oracle Site Studio now maximizes the reusability of all parts of a web site. This includes the structure of the web site, reusing page templates, region templates, element definitions, and other structural components. It also includes being able to reuse all parts of the content and information displayed on the web site. All assets of the web site are now separately managed objects, and each part can be used and then reused to make the web site much easier to maintain.

Page templates now have more reusable parts, and more separately managed parts. Previous releases used layout pages with inline elements. Now with page templates,

placeholders are used on web pages to define where the site content can be edited or replaced.

Each part of a web page can consist of a page template, placeholders, subtemplates, and region templates, each used with a corresponding definition to modify specific implementations of a particular piece.

More Separately Managed Site Assets

To improve reusability, Oracle Site Studio now includes a larger number of separately managed site assets. The following assets are new:

- **Element definitions:** Files that define the editing experience for element types. Specifically, they specify what a contributor can do when editing an element.
- **Region definitions:** Files that define the element definitions available in the region. This then determines what constitutes the editing environment when Contributor is opened for the page containing the region. Region definitions 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.
- **Placeholder definitions:** 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.
- **Page templates:** 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.
- **Region templates:** 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.
- **Subtemplates:** 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.
- **Custom configuration scripts:** JavaScript files that override the default Contributor editor configuration to provide contributors with a customized editing experience.

The new site assets are in addition to the following site assets which existed in previous Oracle Site Studio releases:

- **Contributor data files:** Content files in XML format that are generated by Oracle Site Studio. Contributor data files are edited using the Oracle Site Studio Contributor application.
- **Native documents:** 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.
- **Images:** Graphic files (JPG, GIF, PNG) that are included in content files or page templates (for example, corporate banners or product images).
- **Scripts:** JavaScript files that provide a series of executable commands. Scripts are often used to provide additional functionality to web pages.
- **Cascading style sheets (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.

- **Custom element forms:** Custom elements are customizable HTML files that define custom forms for use in elements (for example, selection forms for specific file types). Oracle Site Studio comes with several predefined custom element forms (in `[CS-Dir]\custom\SiteStudio\support`). These allow the creation of a fully customized editing experience, which is then placed in the Contributor form as if it were a normal Oracle Site Studio element. Custom element forms are also checked into the content server when the Oracle Site Studio component is installed.
- **Validation scripts:** 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).
- **Fragment libraries:** Collections of chunks of code (fragments) that enhance the functionality of an Oracle Site Studio web site (for example, by providing dynamic navigation aids or a standard page footer).
- **Manager configuration settings:** Files that define the functionality that is available in Oracle Site Studio Manager. Manager is a web-based tool that enables designated users (site managers) to modify the structure of a web site.
- **Conversion definitions:** Files that specify the conversion rules for native documents on a web site.
- **Other media:** Any other media files that could be used on a web site, such as Flash animations, video files, audio files, and so on.

New Default Contributor Editor

Oracle Site Studio Contributor now uses FCKeditor as the default editor.

See *Using Oracle Site Studio Contributor* for more information on the Contributor application.

Static and Dynamic Lists as Elements

Static lists and dynamic lists are now elements, and can be more easily implemented and modified using element definitions. In previous versions of Oracle Site Studio, lists were fragments.

More Custom Elements

This release of Oracle Site Studio includes more types of custom elements. Many data-driven web sites can contain values (boolean, integer, and so on) that control the content. With more custom elements, the designer and contributor are able to use more forms of data control. Oracle Site Studio Designer comes with several predefined custom element forms (in `[CS-Dir]\custom\SiteStudio\support`). These forms are also checked into the content server when the Oracle Site Studio component is installed.

Custom Configuration Scripts

Oracle Site Studio now uses custom configuration scripts to customize the editor. The designer can modify the contributor interface to suit specific business needs.

Switch Content Wizard

Changing the content is now handled through a wizard. The wizard makes it easier to use and reuse site assets on the site. You can use the wizard to switch to another contributor data file, native document, or a subtemplate.

Link Wizard

The link wizard has been redesigned for ease of use. It is now easier and faster to create links through the wizard, allowing you different choices of link formats and link targets.

Dialogs More Tightly Integrated in Design Environment

The site asset definition and configuration dialogs are now more tightly integrated in the site design environment. These dialogs can be viewed as tabs in the design area.

1.7 Accessibility Features

Oracle Site Studio complies with Oracle's accessibility guidelines. Accessibility features are provided in a number of areas:

- [Section 1.7.1, "Context Menus"](#)
- [Section 1.7.2, "Standard Windows Keyboard Shortcuts"](#)
- [Section 1.7.3, "Oracle Site Studio Designer-Specific Keyboard Shortcuts"](#)
- [Section 1.7.4, "Fragments and Accessibility"](#)

1.7.1 Context Menus

Oracle Site Studio Designer provides context menus to provide options that are relevant to the current context of the application. For example, with a contribution folder selected, it shows options that are applicable to that folder in that particular context. To show a context menu, right-click in a toolbox or workspace in Designer, or press the menu key on your keyboard.

1.7.2 Standard Windows Keyboard Shortcuts

Oracle Site Studio supports most standard Windows keyboard shortcuts, for example:

- **Ctrl + C**: copy
- **Ctrl + X**: cut
- **Ctrl + V**: paste
- **Delete**: delete
- **F2**: rename
- **Tab**: move to next item
- **Space**: select or clear a checkbox.
- **Alt + F**: opens the File section of the Menu toolbar.

This is just a short list; the set of keyboard commands used in Windows navigation are used in Oracle Site Studio Designer, except where they would not work as expected, as listed in [Section 1.7.3, "Oracle Site Studio Designer-Specific Keyboard Shortcuts."](#)

1.7.3 Oracle Site Studio Designer-Specific Keyboard Shortcuts

The following shortcuts are used to work with the different areas of Oracle Site Studio Designer:

- Workspace screens:
 - **Ctrl + Tab**: moves through the open documents.
 - **Ctrl + PgUp**, **Ctrl + PgDn**: moves through the *Source*, *Design*, and *Preview* tabs on a specific document.
 - **1**: When you are in a site asset that has a columned entry in a form (for example, the WYSIWYG Element) pressing **1** selects the first column.
 - **2**: When you are in a site asset that has a columned entry in a form (for example, the WYSIWYG Element) pressing **2** selects the second column.

The **Tab** key will respond differently when you are working with source code than with other entries. When working with source code, **Tab** will insert a tab, for aligning text. When working with other entry forms, **Tab** will move to the next selection.

- Designer panes:
 - **Ctrl + Tab**: moves between sections in a pane (for example, in the Section Properties pane).
 - **Ctrl + Shift + Tab**: sorts items either alphabetically or in groups, where available (for example, in the Properties Pane).
 - **Enter**: When you are in the Section Properties, press **Enter** to move focus to the text entry. Press **Shift + Enter** to exit from text entry and return focus to the property name.
 - **Shift + Enter**: When the focus is on a section property, but not for text entry in the specific property, then press **Shift + Enter** to click the **Additional Information** button.
- Dialogs:
 - **Esc**: In dialogs that have form entry, such as the Fragment Editor, press **Esc** to enter the form to modify the information.
 - **Shift + Esc**: In dialogs that have form entry, such as the Fragment Editor, press **Shift + Esc** to leave the form and move focus outside the text-entry.

Additionally, dialogs that have list controls, the context menu allows you to add, delete, or move within the list.

- Link Wizard:

The Link Wizard behaves as a Web page would. The arrow keys will not move in the hierarchy, rather you press **Tab** to move to the selected item and **Enter** to select. Additionally, you press **Tab** to move into a radio button group, and **Enter** to select the button.

1.7.4 Fragments and Accessibility

Fragments in Designer have two methods of interaction. There is a Fragment Toolbox, which is a pane in Designer. This toolbox, however, is only accessible through using the mouse.

For complete accessibility, fragments are accessible through a dialog, the Select Fragment dialog (see [Section A.81, "Select Fragment Dialog"](#)). This dialog can be opened from the Menu just as any other dialog, and is navigable using the keyboard.

1.8 System Requirements

The system requirements for the Oracle 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. Both 32 bit and 64 bit systems are supported.
- Access to to an Oracle Content Server 11gR1 or 10gR3 instance running the Oracle Site Studio component.
- Microsoft Internet Explorer 7 or higher. (You can use Microsoft Internet Explorer 7 or higher or Mozilla Firefox 1.0.7 or higher to view the resulting web pages.)

Setting Up Oracle Site Studio Designer on Client Computers

This section explains the process of setting up Oracle Site Studio Designer 11gR1 on client computers running the Microsoft Windows operating system. It covers these topics:

- [Section 2.1, "About Designer"](#)
- [Section 2.2, "System Requirements"](#)
- [Section 2.3, "Installing Designer"](#)
- [Section 2.4, "Uninstalling a Previous Version of Oracle Site Studio Designer"](#)

2.1 About Designer

Oracle Site Studio Designer provides a development environment in which a single individual (the site designer) can create, design, manage, and distribute Oracle 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.

2.2 System Requirements

The system requirements for the Oracle 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 the Oracle Content Server running the Oracle Site Studio component.
- Microsoft Internet Explorer 7 or higher. (You can use Microsoft Internet Explorer 7 or higher, or Mozilla Firefox 1.0.7 or higher to view the resulting web pages.)

2.3 Installing Designer

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

1. Find the Oracle Site Studio Designer installer on the My Downloads page in the content server web interface (under My Content Server).

If you do not have access to an Oracle Content Server 11gR1 instance, contact your system administrator for instructions on where to obtain the client software installer for Oracle Site Studio Designer.

2. After obtaining the client software, run the installer executable.
The installation wizard is launched.
3. On the Welcome screen, click **Next** to start the installation.

Note: If you install Oracle Site Studio Designer 11gR1 on a computer that already has an earlier Designer version on it, the Designer 11gR1 application installs alongside the earlier version. The previous version of Designer is not removed. If you are installing the Designer 11gR1 application and an earlier 11gR1 Designer version is present, you will be prompted to uninstall the earlier version.

Note: You need administrative rights to the computer to install Oracle Site Studio Designer. If you want to install Oracle 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.

2.4 Uninstalling a Previous Version of Oracle Site Studio Designer

If a previous version of Oracle Site Studio Designer is already on your computer, you may want to uninstall it, since Oracle Site Studio Designer 11gR1 will install alongside the older version rather than on top of it.

To completely uninstall Oracle Site Studio Designer, complete these steps:

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

The Oracle Site Studio Designer uninstall wizard is launched.

5. Click **Yes** when asked if you are sure that you want to remove Designer from your computer.

The software is removed from the client computer.

Understanding Oracle Site Studio Web Sites

Oracle Site Studio provides a powerful set of tools that can make designing, building, and maintaining web sites simpler and more efficient. To get the most out of these tools, it is useful to understand some basic concepts about Oracle Site Studio web sites.

This section covers the following topics:

- [Section 3.1, "Managed Web Sites"](#)
- [Section 3.2, "Separation of Site Presentation and Content"](#)
- [Section 3.3, "Site Asset Storage"](#)
- [Section 3.4, "Site Roles"](#)
- [Section 3.5, "Presentation Model"](#)
- [Section 3.6, "Contribution Model"](#)
- [Section 3.7, "Site Object Hierarchy"](#)
- [Section 3.8, "Reusing Site Assets"](#)
- [Section 3.9, "Elements and Element Definitions"](#)
- [Section 3.10, "Region Templates and Region Definitions"](#)
- [Section 3.11, "Placeholders and Placeholder Definitions"](#)
- [Section 3.12, "Page Templates"](#)
- [Section 3.14, "Index Pages and Flex Pages"](#)
- [Section 3.15, "Named Pages"](#)
- [Section 3.13, "Subtemplates"](#)
- [Section 3.16, "Contributor Data Files and Native Documents"](#)
- [Section 3.17, "Fragments"](#)
- [Section 3.18, "Cascading Style Sheets"](#)
- [Section 3.19, "Project Files"](#)

3.1 Managed Web Sites

What differentiates web sites created and managed using Oracle Site Studio from other, conventional sites is that all files associated with the site are stored and managed in a central repository (the content server). This server provides advanced content management features such as library services (check-in, check-out, and the like), versioning, workflow, content conversion, and more.

The files associated with a web site include files that are used to directly control the visual presentation of the site (page templates, cascading style sheets, and the like), and the actual content on the web pages (the "information"). In addition, there are several control and configuration files that work "behind the scenes" to ensure that the site works as it should. Since all files associated with a web site are stored together, they are much easier to control and maintain. For example, it is very easy to back up a complete site or replicate it for deployment. Also, if multiple sites are stored on the content server, then the different assets of the sites can all be maintained in a single location, but be used and reused across all company web sites.

3.2 Separation of Site Presentation and Content

One thing that makes Oracle Site Studio a valuable tool is that it allows web site content to be completely separate from web site presentation. This way, different people can manage and be responsible for the information on a web site without inadvertently affecting the layout, design, or look-and-feel of the site. In addition, the people assigned to manage the site content can make changes as necessary without having to send them to someone else to complete the task. This removes an important bottleneck that exists in many site management scenarios, where all site changes must be handled by a very limited number of site administrators.

With separation of site presentation and content in mind, the files associated with a web site can be divided into three main categories:

- [Section 3.2.1, "Site Presentation Files"](#)
- [Section 3.2.2, "Site Content Files"](#)
- [Section 3.2.3, "Site Control and Configuration Files"](#)

3.2.1 Site Presentation Files

A number of the files associated with an Oracle Site Studio web site are used to define what the site looks like in terms of page layout and formatting. They provide the design framework within which the site content is displayed. Any changes to these files typically affect the entire site (or large portions of it), and they are usually created and managed by dedicated site designers.

Oracle Site Studio uses the following files for site presentation:

- **Page templates:** 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. For more information, see [Section 3.12, "Page Templates."](#)
- **Region templates:** 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. For more information, see [Section 3.10, "Region Templates and Region Definitions."](#)
- **Subtemplates:** 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. For more information, see [Section 3.13, "Subtemplates."](#)
- **Cascading style sheets (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. For more information, see [Section 3.18, "Cascading Style Sheets."](#)

In addition to these files, which directly affect the site presentation, there are also several files that work "behind the scenes" and also have an impact on what an Oracle Site Studio web site looks like. For more information, see [Section 3.2.3, "Site Control and Configuration Files."](#)

3.2.2 Site Content Files

The site content (that is, the actual information on the site) is stored in separately managed files, separate from the presentation context in which they appear. This enables them to be managed separately and reused within a web site, or even between web sites (providing these sites are all managed using the same content server).

Oracle Site Studio uses the following files for site content:

- **Contributor data files:** Content files in XML format that are generated by Oracle Site Studio. Contributor data files are edited using the Oracle Site Studio Contributor application. For more information, see [Section 3.16, "Contributor Data Files and Native Documents."](#)
- **Native documents:** 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. For more information, see [Section 3.16, "Contributor Data Files and Native Documents."](#)
- **Images:** Graphic files (JPG, GIF, PNG) that are included in content files or page templates (for example, corporate banners or product images).
- **Other media:** Any other media files that could be used on a web site, such as Flash animations, video files, audio files, and so on.

3.2.3 Site Control and Configuration Files

In addition to the files that directly affect site presentation, there are several files that also have an impact on what an Oracle Site Studio web site looks like. You generally do not see them visually represented on the web site, but they do work "behind the scenes" to make sure the site looks and functions as intended.

Oracle Site Studio uses the following files for site control and configuration:

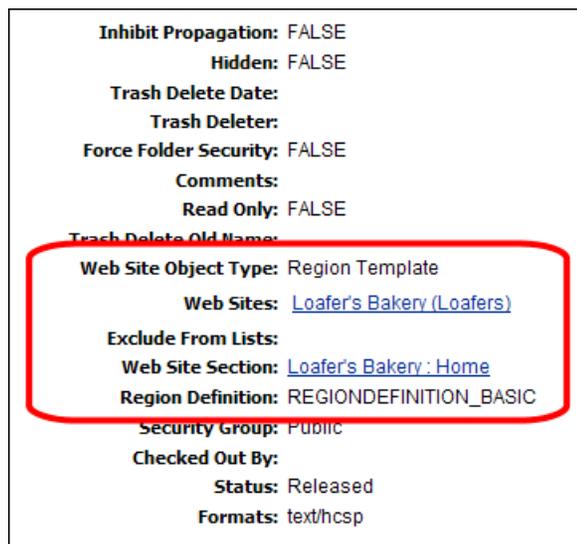
- **Element definitions:** Files that define the editing experience for element types. Specifically, they specify what a contributor can do when editing an element. For more information, see [Section 3.9, "Elements and Element Definitions."](#)
- **Region definitions:** Files that define the type of 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. For more information, see [Section 3.10, "Region Templates and Region Definitions."](#)
- **Placeholder definitions:** 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. For more information, see [Section 3.11, "Placeholders and Placeholder Definitions."](#)
- **Scripts:** 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.

- **Custom configuration scripts:** JavaScript files that override the default Contributor editor configuration to provide contributors with a customized editing experience.
- **Custom element forms:** HTML files that define custom forms for use in elements (for example, selection forms for specific file types). Oracle Site Studio comes with several predefined custom element forms (in `[CS-Dir]\custom\SiteStudio\support`). (These forms are also checked into the content server when the Oracle Site Studio component is installed.)
- **Validation scripts:** 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).
- **Fragment libraries:** Collections of chunks of code (fragments) that enhance the functionality of an Oracle Site Studio web site (for example, by providing dynamic navigation aids or a standard page footer).
- **Manager configuration settings:** Files that define the functionality that is available in Oracle Site Studio Manager. Manager is a web-based tool that allows designated users (site managers) to modify the structure of a web site.
- **Conversion definitions:** Files that specify the conversion rules for native documents on a web site.

3.3 Site Asset Storage

All files associated with an Oracle Site Studio web site are stored and managed using Oracle Content Server. A number of custom metadata fields specific to Oracle Site Studio are used to specify where and how the files are used.

Figure 3–1 Oracle Site Studio Metadata Fields on Content Information Page of Site Asset



Screenshot of part of a content information page, with the Oracle Site Studio specific metadata fields highlighted. The fields are called Web Site Object Type, Web Sites, Exclude From Lists, Web Site Section, and Region Definition, and they are explained in more detail in the text that follows this image.

Oracle Site Studio uses the following metadata fields for all site-related files:

- **Web Site Object Type:** Specifies the type of file for the web site (for example, "Data File," "Stylesheet," "Region Template," and "Placeholder Definition").
- **Web Sites:** Lists the web site(s) that the file is associated with. This means that the file can be used on the listed site(s), although it is not necessary. A file may be associated with multiple web sites, which means it can be reused between sites. This makes multi-site management more efficient.
- **Exclude From Lists:** Lists the web site(s) for which a contributor has specified that a particular content file (a contributor data file or native document) should not display in dynamic lists on the web site.
- **Web Site Section:** Specifies where a content file is displayed by default on a web site (unless a target section is explicitly specified in the original hyperlink).
- **Region Definition:** Specifies the region definition that a region template or content file (contributor data file or native document) is associated with. This determines how the file opens on the site and what contributors are allowed to do. A region template and content file can be associated with only one region definition, but a region definition may have many region templates and content files associated with it.

Please note that some fields are automatically set as the file is created and checked into the content server. Also, not all metadata fields are used for all site assets, since they may not be relevant. For example, the `xRegionDefinition` metadata field is not used for page template, since these are not associated with region definitions.

You can change the values of these fields on the content information page of a file, but be carefully when doing that, since this may affect the way the file is used on a site.

3.4 Site Roles

When the various roles for web site creation and management within the organization are determined, each can focus on specific tasks in making the web site work.

Designers would focus on how the web site looks: the structure of the pages, the way the pages are laid out, the imagery, and the corporate identity. This document focuses on the role of designers.

Contributors are then able to place the content on the page without having to code the pages. More importantly, the contributors can update and edit the content without affecting how the page is displayed, and without having to make multiple changes to multiple areas of the web site. The designer, for the most part, does not control the content, and contributors, for the most part, do not control how the content opens on the site. The contributor role is discussed in more detail in *Using Oracle Site Studio Contributor*.

Managers are able to reorganize the site navigation and hierarchy. They can add or remove sections to the site without using Oracle Site Studio Designer by using a web-based tool that the site designer makes available to them. The manager role is discussed in more detail in *Managing Oracle Site Studio*.

In addition to these three Oracle Site Studio roles, there may also be separate site **administrators**, who could take on responsibility for assigning web addresses to the site, backing up the site, replicating the site, and so on. All these administrative tasks are performed on the content server. The site administration tasks are discussed in more detail in *Managing Oracle Site Studio*.

3.5 Presentation Model

Oracle Site Studio completely separates the presentation layer from the content layer of a web site, as shown in [Figure 3-2](#).

Figure 3-2 Presentation Model for Oracle Site Studio Web Sites

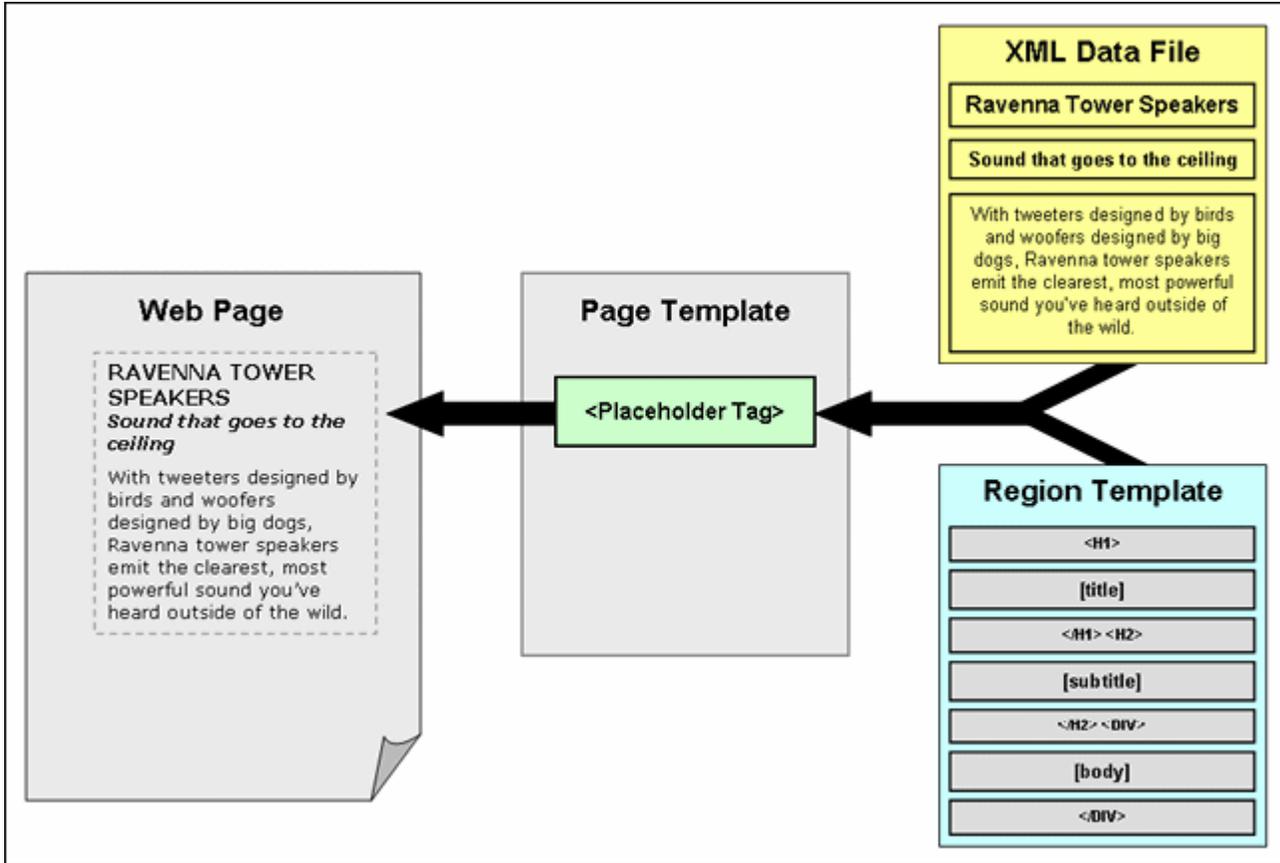


Diagram of the presentation model for Oracle Site Studio web sites. There is a strict separation of content and presentation, which is described in more detail in the text surrounding this image.

Page templates are used to define the site framework within which content is displayed. They contain standard HTML layout and formatting code, along with Oracle Site Studio tags to specify where fragments and/or placeholders go. Placeholders specify where the contribution regions (that is, editable areas) are on the page. Please note that they do not specify anything about what goes in these regions, both in terms of content and visual presentation. That is handled by region templates (with their associated region definitions).

Region templates define the layout and look-and-feel of the data in a contribution region (marked on a page template using a placeholder tag). They are separately managed site assets, which makes them easy to reuse within web sites, or even between web sites. (Please note that in Oracle Site Studio releases prior to 10gR4 region presentation was not separately managed, but included within page templates, or 'layout pages' as they were called in these previous releases.)

The content of contribution regions is stored in data files, which are also separately managed site assets. When it is time to generate a web page, Oracle Site Studio looks at the placeholder on the page template, takes its associated region template and data file, and merges these two to create HTML code that is inserted into the page template at the position of the placeholder tag. This creates the final web page, which all content in place, presented and formatted in accordance with the site and page settings.

3.6 Contribution Model

As with the presentation model, the contribution side of Oracle Site Studio web sites separates site content from presentation, as shown in Figure 3-3.

Figure 3-3 Contribution Model for Oracle Site Studio Web Sites

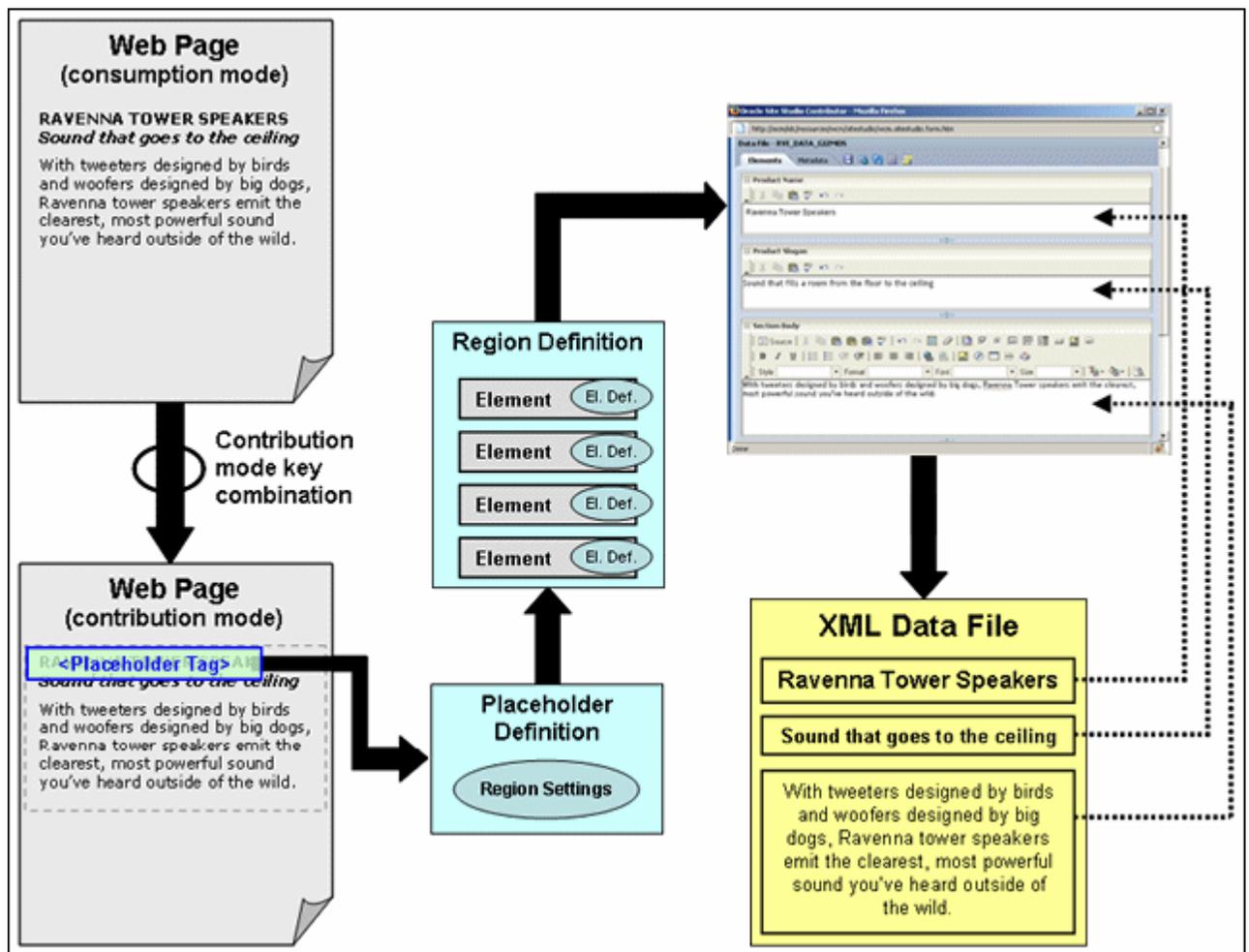


Diagram of the contribution model for Oracle Site Studio web sites. There is a strict separation of content and presentation, which is described in more detail in the text that follows this image.

When site contributors decide that they want to edit the contents of a web page, they press a key combination to reload the page in contribution mode. (The default key combination is Ctrl+Shift+F5, but this may be modified.)

After a page is in contribution mode, all contribution regions (that is, editable areas) on a web page are marked and underlying code has been added to the page to identify what placeholder is associated with the contribution region. Based on this information, Oracle Site Studio can establish what placeholder definition and region definition are used for the placeholder. The region definition identifies the structure of the content in the contribution region in terms of its constituent data segments (elements). Each element has an element definition, which defines what editing options are available to contributors for the element.

When the contributor decides to edit the content in a contribution region, its associated contributor data file is checked out of the content server. The structure of the data file matches that of its associated region definition in terms of number and types of elements. The data is loaded from the data file and presented in the Contributor editor. Each element in the data file is presented as an editable element in the editor, with the editing features as defined in the element definition for the element type.

When the contributor is done editing and clicks the **Save** icon in the Contributor editor, the data file is updated and checked into the content server again. The web page is then updated in accordance with the site update schedule.

Please note that the Contributor editor always shows *all* elements in the region definition (and hence, data file) associated with a contribution region, even if they are not used in that particular region. The other elements may be used elsewhere on the web site, so editing that information may affect other pages on the site.

3.7 Site Object Hierarchy

[Figure 3-4](#) shows the hierarchy of site objects that are used to create and manage an Oracle Site Studio web site.

Figure 3–4 Object Hierarchy of Web Sites

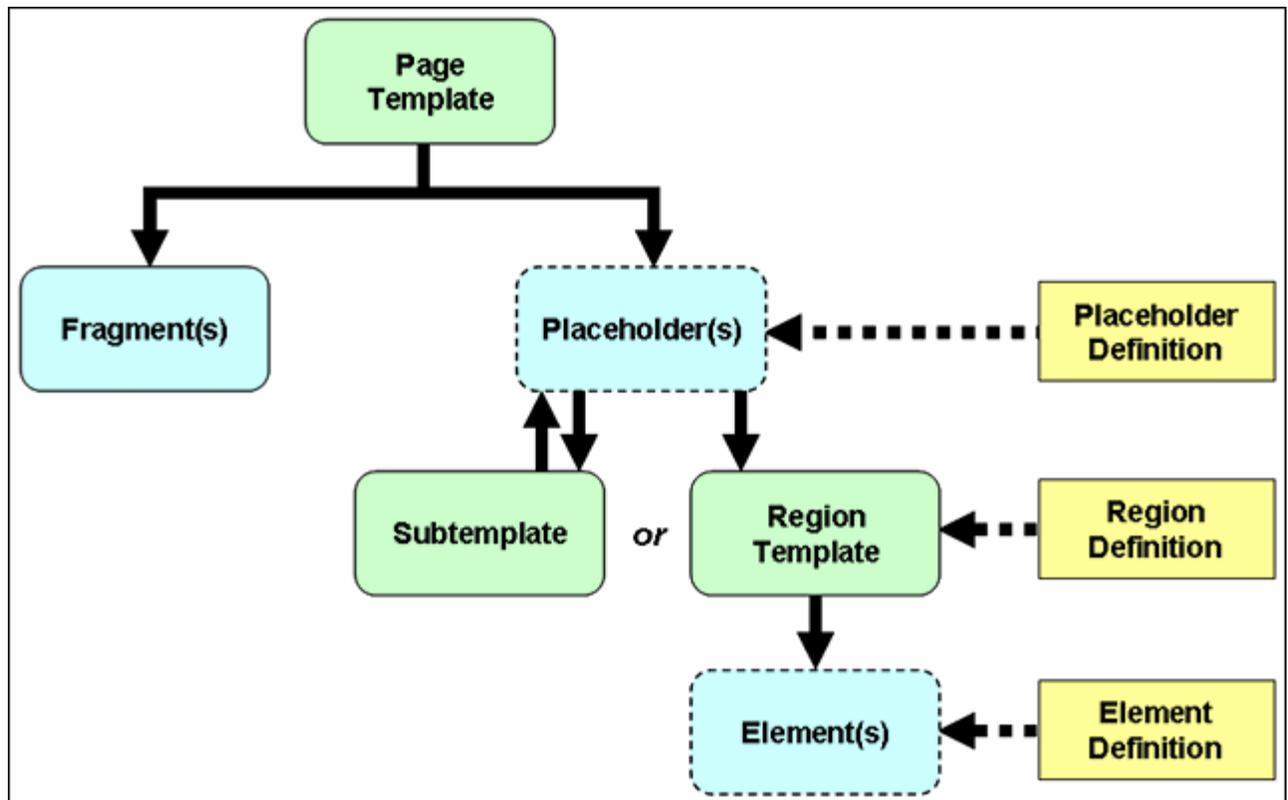


Diagram that shows the hierarchy of objects within an Oracle Site Studio web site. It is described further in the text that follows this image.

Page templates are at the top of the hierarchy. They provide the framework for the pages in a web site within which the site content is displayed. In addition to standard HTML layout and formatting code, they contain site-wide images and other assets, and tags for fragments and/or placeholders. Page templates are stored and managed on the content server. For more information, see [Section 3.12, "Page Templates."](#)

Fragments are chunks of code that can be added to a page template to enhance its functionality. Oracle Site Studio comes with several predefined fragments (for example, for dynamic navigation aids), but you can also create your own fragments. A page template may contain multiple fragments. Fragments are stored in fragment libraries. For more information, see [Section 3.17, "Fragments."](#)

A **placeholder** is no more than an insertion point (a tag) on a page template to identify where there is a contribution region (that is, editable area) on the web page. What that contribution region contains and what it looks like is defined using region templates and region definitions. A page template may contain multiple placeholders. There are no files associated with placeholders; that is, there are no "placeholder files" on the content server. Placeholders are controlled by **placeholder definitions**, which specify what content can go in the contribution region and how it is displayed, as well the actions available to contributors (for example, switching content or modifying metadata). A placeholder contains either one subtemplate or one region template. For more information, see [Section 3.11, "Placeholders and Placeholder Definitions."](#)

A **subtemplate** is a partial HTML file (that is, without a head and body section) that provides a mechanism to divide a placeholder on a page template into further smaller,

reusable areas with their own placeholder(s). There is a circular relationship between placeholders and subtemplate; that is, a placeholder may contain a subtemplate, which, in turn, may include one or more placeholders. Subtemplates are stored and managed on the content server. For more information, see [Section 3.13, "Subtemplates."](#)

A **region template** is a partial HTML file (that is, without a head and body section) that defines the layout and look-and-feel of the data in a contribution region (marked on a page template using a placeholder tag). Region templates are controlled by **region definitions**, which define what kind of content can go in the region template. They also specify the content creation and switching options available to contributors for the contribution region, and set default metadata for content files associated with the region. Both region templates and region definitions are stored and managed as separate assets on the content server. A region template may have one or more references to elements. For more information, see [Section 3.10, "Region Templates and Region Definitions."](#)

Elements are the smallest chunks of reusable information in an Oracle Site Studio web site. They are referenced in region templates, which causes their data to be pulled into the region template using the layout and presentation defined in the template. A region template may contain multiple element references. There are no files associated with elements as such; that is, there are no "element files" on the content server. Groups of elements are arranged in region definitions, which specify site content types. Elements are controlled by **element definitions**, which specify the editing experience available to contributors for an element type. Specifically, they set the available editing features in the Contributor editor when a contributor is editing elements in a contributor data file. For more information, see [Section 3.9, "Elements and Element Definitions."](#)

[Figure 3–5](#) shows an example of a site object hierarchy.

Figure 3–5 Example of Site Object Hierarchy

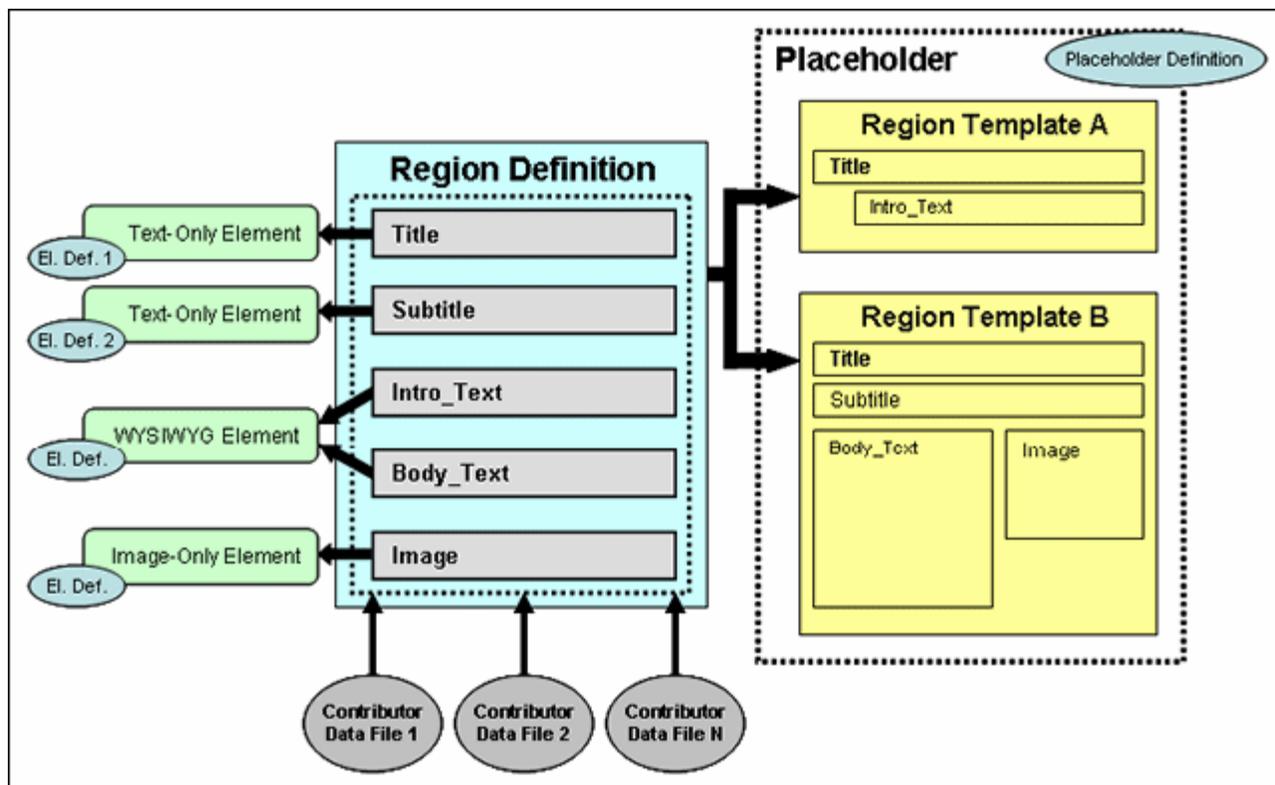


Diagram that shows an example of a site object hierarchy, which is described in more detail in the text surrounding this image.

In Figure 3–5, there is a placeholder (that is, an editable contribution region on a page template) that has two region templates available to it (as specified in the placeholder definition). Region template A shows a limited data view, with just the title and a brief introductory text. Region template B presents a more elaborate data view, with the title, subtitle, body text, and an image. Either can be used for the placeholder, depending on the site context. Both region templates are associated with a region definition that has elements for each of the reusable chunks of information (Title, Subtitle, Intro_Text, Body_Text, and Image). Each of these elements is associated with an element definition.

The Title and Subtitle elements are of the same type (text only), but they have different element definitions, which means that the editing features available to contributors are different. The Intro_Text and Body_Text elements are both WYSIWYG elements, which typically means that contributors have a broad array of editing options available to them when editing these elements (for example, the ability to add tables or use advanced text formatting). The editing experience for contributors is the same for these elements.

One or more contributor data files are associated with the region definition, and ultimately with the contribution region. Their structure matches that of the region definition. They contain the same elements: Title, Subtitle, Intro_Text, Body_Text, and Image. When a contributor decides to edit the content in a contribution region, its associated contributor data file is loaded into the Contributor editor, which provides one editing area for each element in the data file. The editing features available in each area are set by the element definition.

3.8 Reusing Site Assets

One of Oracle Site Studio's most useful and powerful features is the ability to reuse site assets within a web site and even across multiple sites (providing these sites are all managed on the same content server). When an asset is changed once, it is changed everywhere that asset is used. It is no longer necessary to keep track of all instances of a piece of data to ensure that all of the web pages are updated. This applies both to files associated with site presentation and site content (see Section 3.2, "Separation of Site Presentation and Content"). Page templates, region templates, elements, and the like are most efficiently used when they are used multiple times. Similarly, the same content files can be displayed in different locations on a site, either completely or partially (different segments), to suit the context.

Figure 3-6 and Figure 3-7 show an example of site content being reused. In Figure 3-6, you can see a list of items consisting of a title, image, and subtitle. This information is taken from separate contributor data files. Each of these list items could be hyperlinked to open the full page as shown in Figure 3-7. In fact, this is the type of design that you see on many sites: a "teaser" of information opens on a landing page, and the full information is shown on a subsequent page when you click a teaser link.

Figure 3-6 Web Page With "Teaser" Content

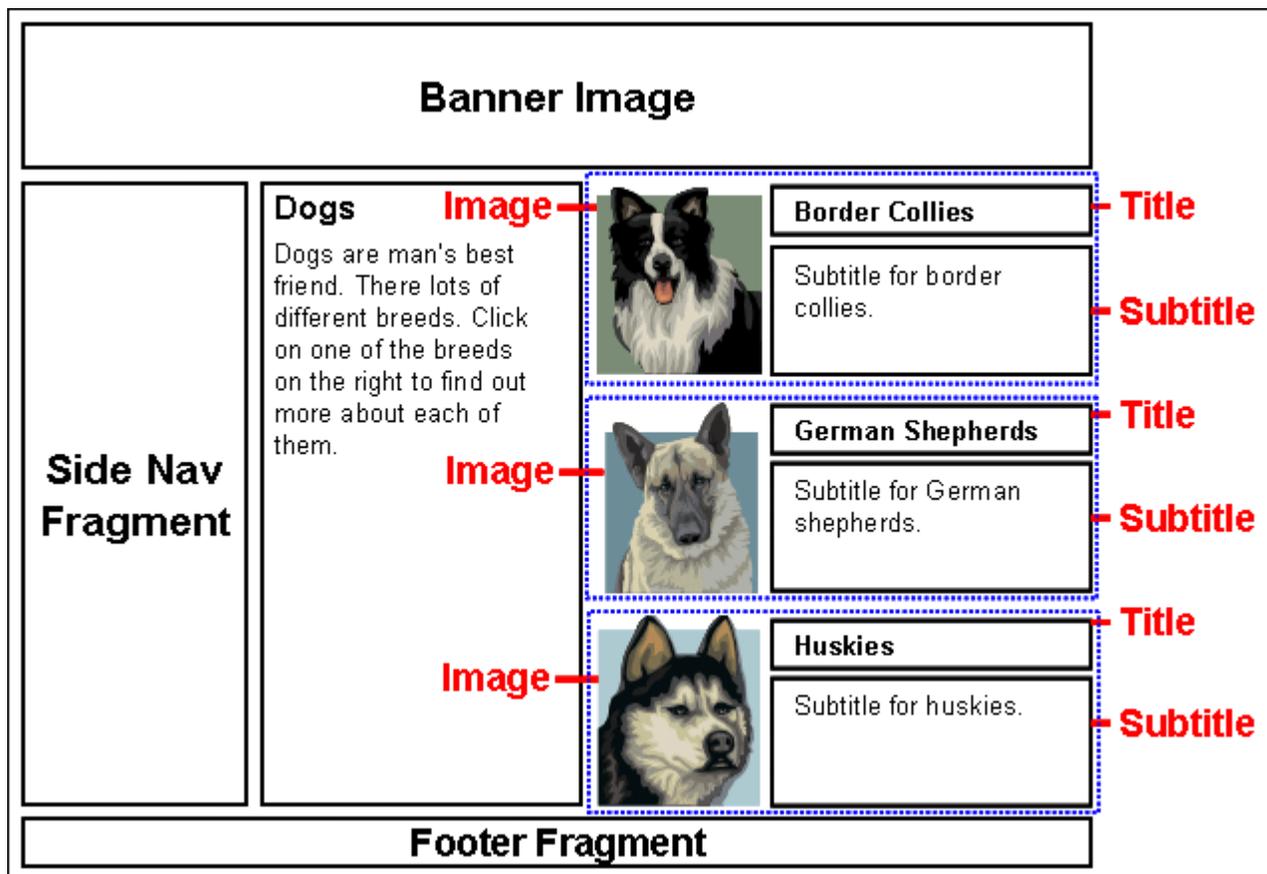


Diagram of a web page with a list of items, each of which consists of a title, a subtitle, and an image for three different dog breeds (Border Collies, German Shepherds, and Huskies).

Figure 3-7 Web Page With Full Content

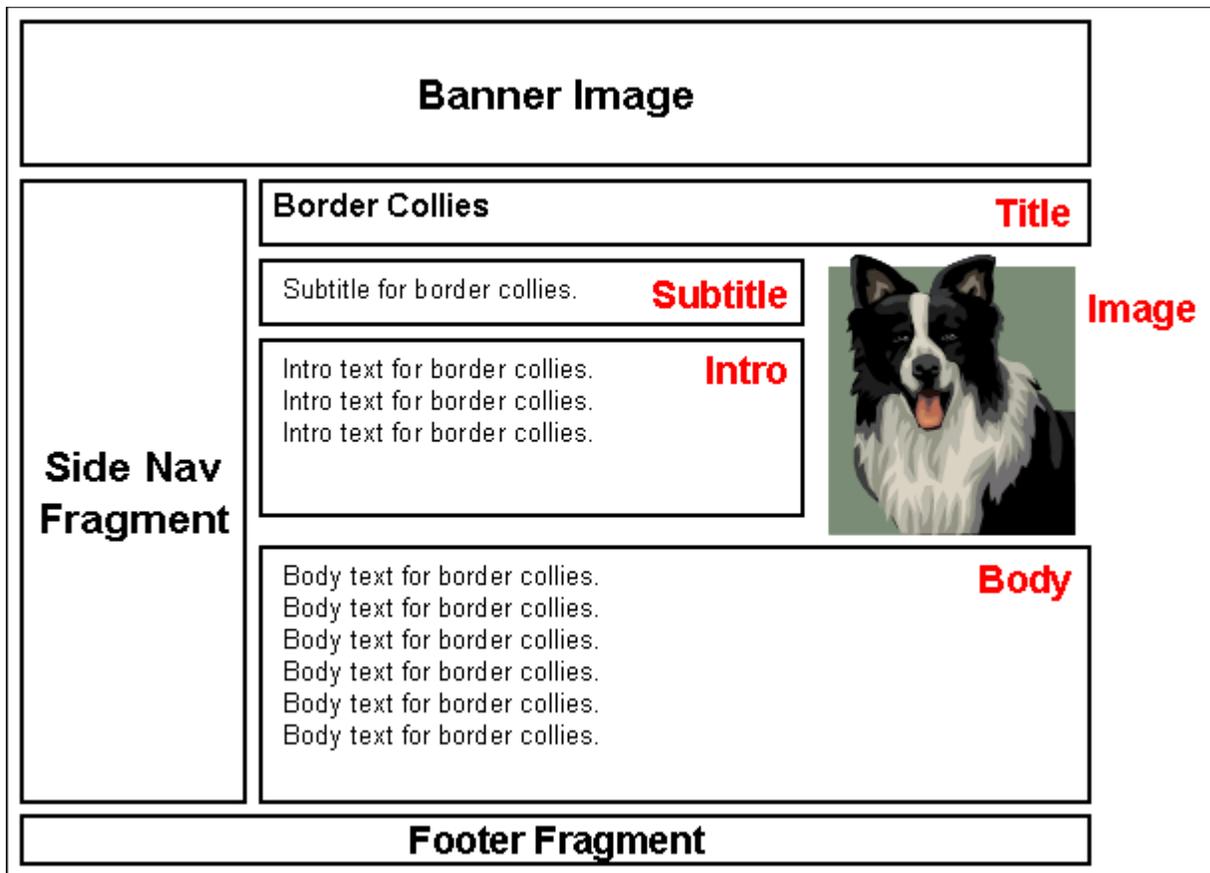


Diagram of a web page that shows the full page on a dog breed from Figure 3-6. It uses the same contributor data file as in Figure 3-6, but includes introductory text and body text, in addition to the title, subtitle, and image.

The web page in Figure 3-7 displays many of the same pieces of information. That is because the same data file is being used on both the region template in Figure 3-6 and the page in Figure 3-7. The elements are the same as well. An image-only element would be used to present the image, for instance, and a text-only element to present the title, with WYSIWYG possibly being the element type used to display the remaining information. There are few elements, typically, necessary in creating a web site, as one WYSIWYG element can be used anywhere on the web site where you may want that style of editing for the contributor.

Because site assets are intended to be reused across the web site, it is particularly important that the web site is fully planned out before the designer creates anything in the web site. For more information, see Chapter 4, "Efficient Web Site Planning."

3.9 Elements and Element Definitions

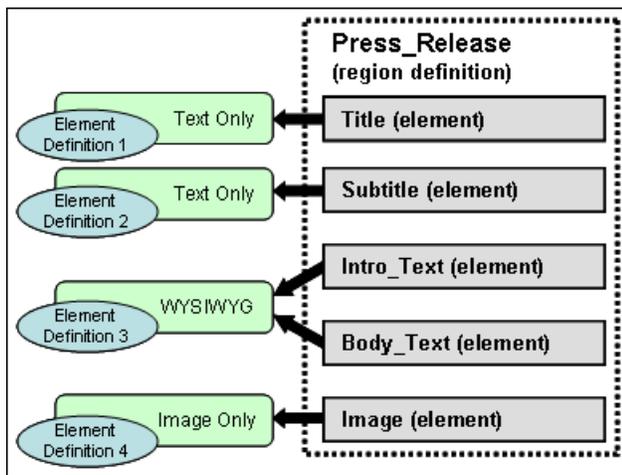
Elements are the smallest chunks of reusable information in an Oracle Site Studio web site (for example, a title, a product image, or the body text of a press release). Since element data can be reused within (or even between) web sites, you, as the site designer, should carefully consider how the site content should be broken up into

segments. This may take some work up front, but maximizes reusability of site content and also makes managing the site content more efficient in the long run.

Each defined element is of a particular type: WYSIWYG, text only, image only, static list, dynamic list, or custom. These types characterize what the element content consists of, and, through element definitions, what editing options are available to contributors. For example, the title of a press release could be set up as a text-only element (which typically provides only limited editing options to contributors), whereas the actual press release text could be a WYSIWYG element (which typically gives contributors much more editing control, such as the capability to add images or tables).

Elements are controlled by **element definitions**. An element is basically an instantiation of its associated element definition, which specifies the editing experience available to contributors for the element on the web page (specifically, what editing features are available in the Contributor editor). Element definitions are individually managed site assets, which means that they can be reused within a web site, or even across web sites (providing all sites are managed on the same content server). As shown in [Figure 3-8](#), elements of the same type (Title and Subtitle) may have different element definitions associated with them to provide different editing environments for contributors depending on the context in which the element is used. Similarly, multiple elements (Intro_Text and Body_Text) may share the same element definition, providing the same editing experience to contributors for each of the elements.

Figure 3-8 Elements and Element Definitions



A diagram showing five elements, called Title, Subtitle, Intro_Text, Body_Text, and Image, with their associated element types and element definitions. The Title and Subtitle elements are both text-only elements, whereas the Intro_Text and Body_Text elements are both WYSIWYG elements. The Image element is an image-only element. All elements are contained in a region definition called "Press_Release." The diagram is explained in more detail in the surrounding text.

Individual elements are not separately managed site assets; that is, there are no "element files" on the content server. (Please note that there are element definition files.) Groups of elements are arranged in region definitions, which specify site content types. As shown in [Figure 3-8](#), there could be a region definition called "Press_Release," which consists of the elements Title, Subtitle, Intro_Text, Body_Text, and Image. A region definition could thus be thought of as a 'content class'.

Element data is stored in contributor data files associated with region definitions. Each contributor data file contains an instance of each element from its associated region definition. In the example of [Figure 3–8](#), the contributor data files would have five elements called Title, Subtitle, Intro_Text, Body_Text, and Image. Contributors can edit the element data in the contributor data files using the editing features available to them for each element, as set in the associated element definitions.

3.10 Region Templates and Region Definitions

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, as shown in [Figure 3–8](#) on page 3-14, 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; see [Section 3.9, "Elements and 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.

Region templates are partial HTML files that define the layout and look-and-feel of the data in contribution regions within web pages. They are partial HTML files in that they do not have a head and body section. This allows them to be inserted into other HTML code as the web pages are generated for the Oracle Site Studio site.

Region templates consist of standard HTML layout and formatting code, along with Oracle Site Studio tags to specify where elements (from contributor data files) or dynamic conversions (of native documents) are placed. Some elements from contributor data files may be displayed in some region templates, but not in others, which allows the information to be reused across different pages (as shown in [Figure 3–5](#) on page 3-11).

You, as the site designer, will probably create region templates more than any other site asset. Region templates allow you to present the information in contributor data files or native documents differently in various contexts of the web site. As with elements, it is worthwhile spending time considering how the information on your site should be presented through region templates. Judicious use of region templates maximizes the reusability of site content, and also makes site content management more efficient.

In short, region definitions specify what contribution regions on a web page contain, whereas region templates define what contribution regions look like. In other words, region definitions specify the structure (and attributes) of site content, and region templates define the visual presentation of that content on a web page.

There can be multiple region templates for each region definition. This allows site content to be displayed differently depending on the context within the site. If there are multiple region templates for a region definition, then the default region template is used unless a different one was specifically set to be used.

Region templates can be used to display information in multiple places in different layouts, while using data from the same data files. A common example of this is a "teaser" list of items showing, for example, a title, a brief subtitle, and a small image. An example of this is shown in [Figure 3-9](#), which represents a region template (with sample content) that shows these three elements arranged with the image on the left and the title and subtitle on the right.

Figure 3-9 Region Template With Limited Element Set From Region Definition

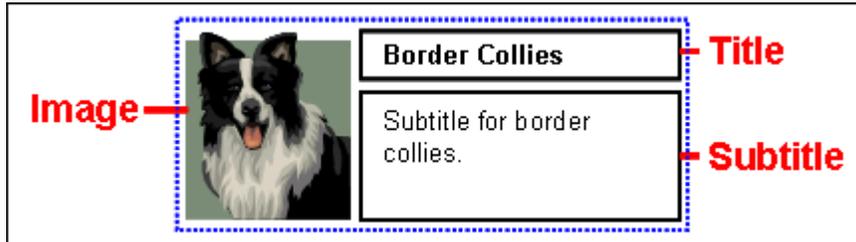


Diagram of a region template which shows a limited element set from a region definition. It is described in more detail in the text surrounding this image.

The content in [Figure 3-9](#) could be hyperlinked to open a new page with a different region template that shows the same data, but now with more elements from the data file. [Figure 3-10](#) shows an example of such a region template (with sample content), with a title at the top and a subtitle, introductory text, and body text below it. The region template also includes the image (although sized and positioned differently). All content is taken from the same data file as used in [Figure 3-9](#).

Figure 3-10 Region Template With Full Element Set From Region Definition

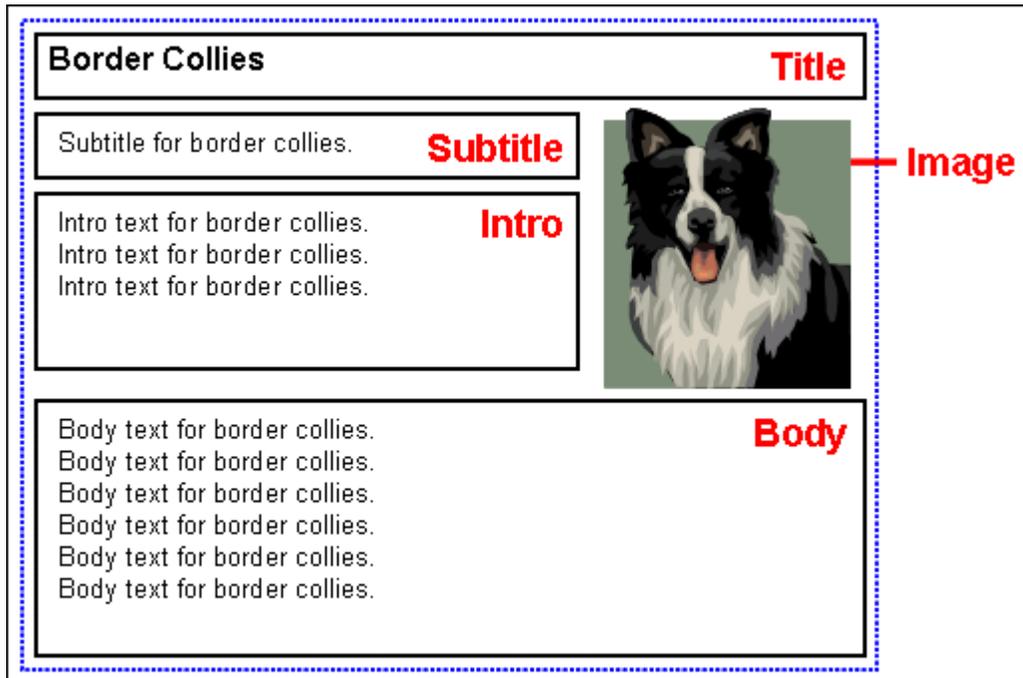


Diagram of a region template which shows the full element set from a region definition. It is described in more detail in the text surrounding this image.

In the case of [Figure 3–9](#) and [Figure 3–10](#), the contribution regions could be set up to allow contributors to edit the data on either page, but they would be editing the same data file and any changes would be reflected in all places where the data file is used. Please note that if contributors edited the data file from the region template showing the limited element set ([Figure 3–9](#)), they would see *all* elements of the data file in the Contributor editor, even though not all elements are being used in that particular context.

3.11 Placeholders and Placeholder Definitions

A **placeholder** is no more than an insertion point (a tag) on a page template (see [Section 3.12, "Page Templates"](#)) to identify where there is a contribution region (that is, editable area) on the web page. [Figure 3–11](#) shows a marker for a placeholder called "Default" in design view, with table layout being used for positioning objects on the page. In source view, a placeholder is represented by a simple tag:

```
<!--$wcmPlaceholder("Name")-->
```

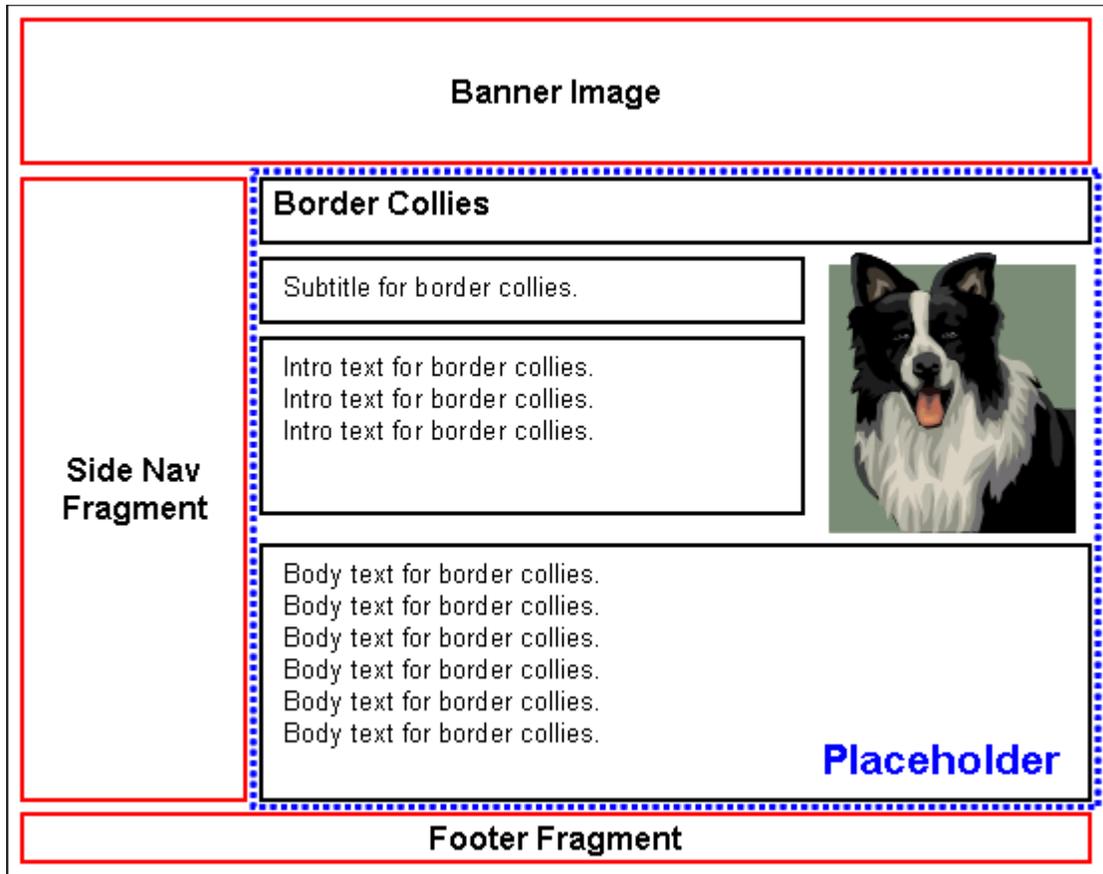
Figure 3–11 Placeholder Marker in Design View



Placeholder marker in design view, with table layout being used for positioning objects on the page. The image shows a three-row table, with an image in the top row, a SiteMapTree fragment and a marker for a placeholder called "Default" in the second row, and a blank third row.

What the contribution region identified by the placeholder contains, and how it looks on the site, is defined using region templates and region definitions (see [Section 3.10, "Region Templates and Region Definitions,"](#)). A page template may contain multiple placeholders, each of which representing a contribution region on the page. There are no files associated with placeholders; that is, there are no "placeholder files" on the content server. [Figure 3–12](#) shows a representation of a web page that contains one placeholder, marked with a dotted line (with sample content in place).

Figure 3–12 Placeholder on Web Page (With Sample Content)



Representation of a web page with one banner image at the top, a navigation fragment on the left, a footer fragment at the bottom, and one placeholder (that is, contribution region) in the center. The contribution region contains sample content, as defined by region templates and region definitions.

When you insert a placeholder in a page template, all you are doing is marking a named position in the template where content may be inserted. To control how content is handled at that position, you must associate the placeholder name with a **placeholder definition**. Placeholder definitions specify what content can go in the contribution region and how it is displayed, and the actions available to contributors. For example, a placeholder may be set up to allow contributors to update the metadata of content displayed in the contribution region, or they may be allowed to switch the content of contribution regions. (Please note that region definitions control what kind of content contributors can switch to.)

Associating a placeholder with a placeholder definition (also called 'mapping') can be done in several ways:

- At the site level (using global mappings): You can set up default placeholder mappings that apply to the entire site unless any section-level or placeholder-level overrides are specified. You do this in Designer by opening the Tools menu and then choosing Define Placeholder Definition Mappings. This enables you to associate placeholder names with placeholder definitions for primary and secondary pages. The placeholder names are subsequently referenced in page

templates. Placeholder names may be used on multiple templates in multiple sections and the mappings still apply.

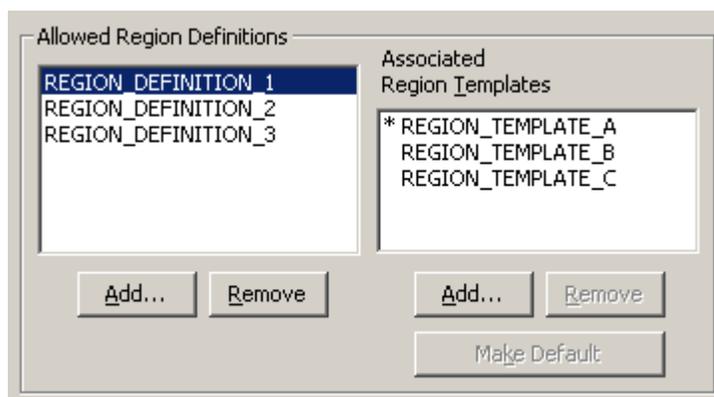
- At the section level (using a section property): You can also set a specific placeholder definition mapping for the primary and/or secondary page of a site section. If you establish the mapping in this way, then it replaces any mapping that may have existed at the site-wide (global) level. Please note that the section where you do this now uses different mapping from any other section where the same placeholder is used.
- At the placeholder level (using a parameter in the placeholder tag): You can also set a specific placeholder definition for a specific placeholder. You do this by adding the `placeholderDefinitionDocName=[NAME]` to the placeholder tag (in source view). If you establish the mapping here, then this overrides all section-level and site-wide (global) mappings. It also means that this template uses this definition everywhere it is used. The template then always uses the specified definition regardless of where in the site it is used. Also note that the only way to change this is to modify the template in source view. This would be considered the least flexible way of specifying a definition mapping as it is hard-coded, but this method can be used if desired.
- As a "catch-all" (using web site properties): You can set up a default placeholder definition that is used if none of the three other methods above apply. You do this by setting the Default Placeholder Definition property in the web site properties category of a site.

You can also set up a default placeholder definition, which serves as a "catch-all" placeholder (if no other applies).

Placeholder definitions specify what content can go in a contribution region (as marked by a placeholder tag) and how this content is displayed, as well the actions available to contributors. For example, a placeholder may be set up to allow contributors to update the metadata of content displayed in the contribution region, or they may be allowed to switch the content of contribution regions. (Please note that region definitions control what kind of content contributors can switch to.)

Placeholder definitions also specify what region definitions, region templates, and subtemplates are available for the associated placeholder (that is, contribution region) on the web page. Take, for example, the placeholder definition shown in [Figure 3–13](#), which allows three region definitions: `REGION_DEFINITION_1`, `REGION_DEFINITION_2`, and `REGION_DEFINITION_3`.

Figure 3–13 *Region Definitions and Templates Associated With a Placeholder Definition*



Partial screenshot of the placeholder definition dialog, which shows the list of allowed region definitions and the associated region templates for the currently selected region definition. This is described in more detail in the text surrounding this image.

In [Figure 3–13](#), region definition 1 has three associated region templates: A, B, and C, with region template A being the default one (indicated by the asterisk). This means that region template A is applied to all content files (contributor data files or native documents) associated with region definition 1 unless a different region template was specifically set. You generally associate a region definition with a content file when you create that file, although you can always change this association on the content information page (see [Section 3.3, "Site Asset Storage"](#)).

3.12 Page Templates

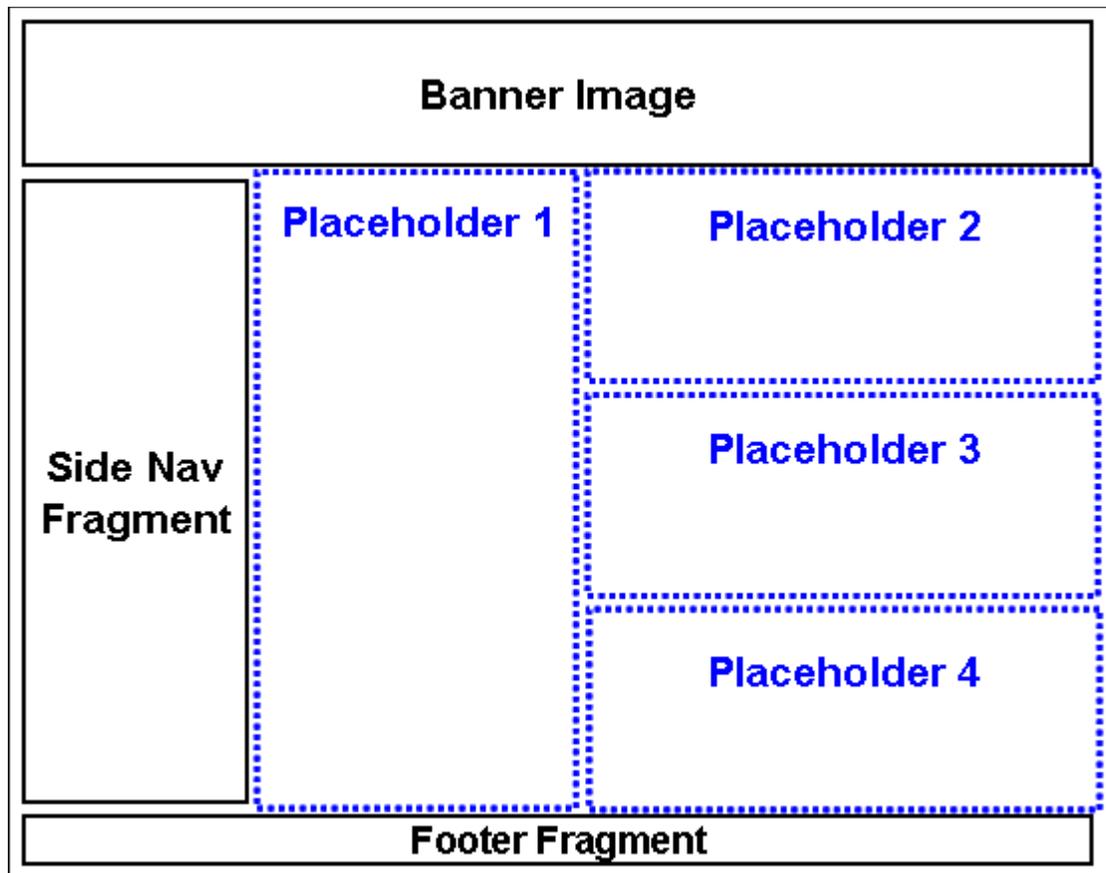
Page templates are fully-formed HTML files that define the layout and high-level look-and-feel of web pages, including the placement of contribution regions (that is, editable areas on the page), navigation aids (in the form of fragments) and site-wide images (banners and the like). They provide the framework within which site content is displayed.

Page templates consist of standard HTML layout and formatting code, along with Oracle Site Studio tags to specify where fragments and/or placeholders go. As such, they are typically light-weight in that they only contain high-level references to where contribution regions go on the page; they do not specify anything about what goes in these regions, both in terms of content and visual presentation. (That is handled by region definitions and templates; see [Section 3.10, "Region Templates and Region Definitions."](#)) Page templates typically include site-wide graphics, such as corporate banners or page layout images, and recurring, non-editable content, such as navigation aids or standard page content (footers and the like).

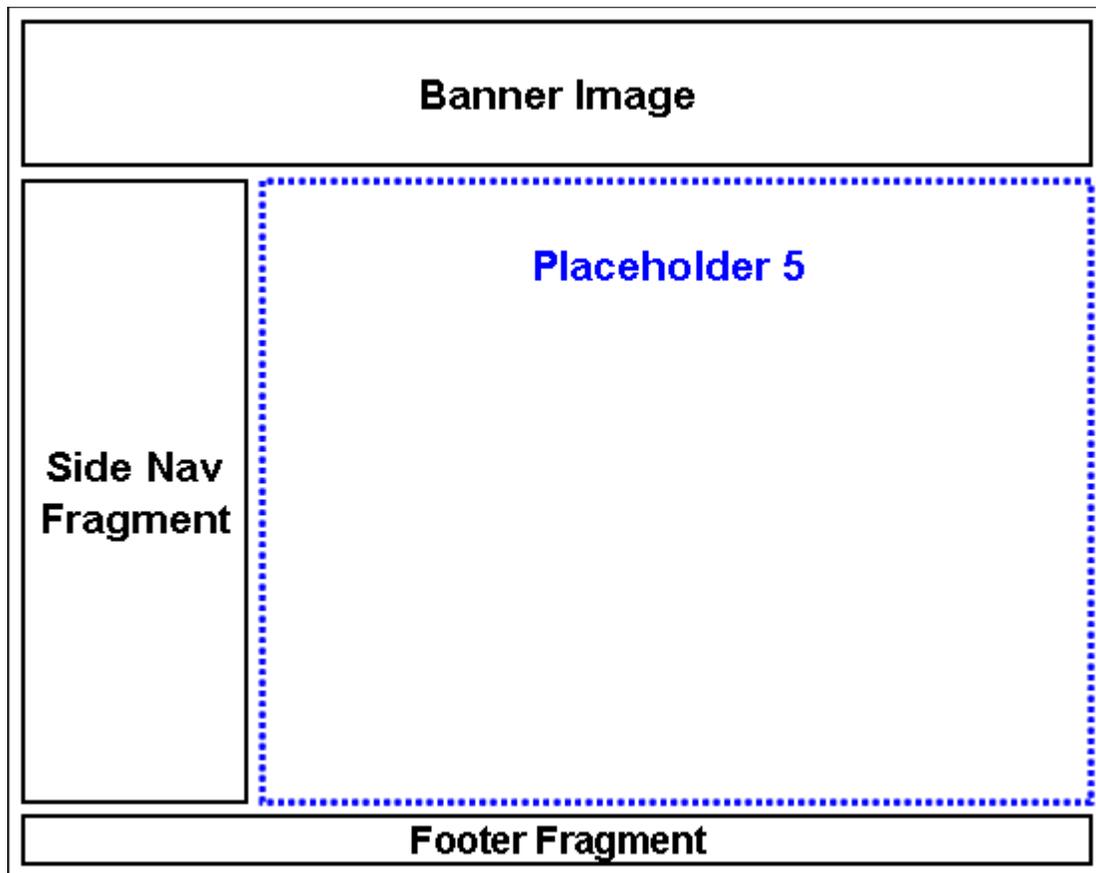
The number of page templates required for a web site depends on the site complexity, but usually only a very limited number should be necessary. For example, it could be sufficient to use one page template for the home page, and another page template for all other pages. The variations of content displayed on the various web site pages could then be handled using region templates (and region definitions) within the placeholders on the page templates. With fewer page templates, a web site is easier to maintain. This does require more work up front; that is, more thought must be put into the design of a web site, how the materials are laid out, and what information appears in which sections of the web site, and so forth. This also means that the site designer should carefully consider what information contributors are allowed to change, and what should remain fixed.

[Figure 3–14](#) shows a generic page template that is often used for the homepage of a web site. It shows a banner image at the top, a navigation fragment on the left, a footer fragment at the bottom, and several small placeholders in the center, each of which representing some "teaser" content. (Please note that placeholders do not delineate any page areas. They are no more than insertion points that specify the location of a contribution region. What goes in these regions and how it is presented is handled by region templates and region definitions.) Site visitors can then click this "teaser" content to see the content in full. This full content could be displayed in a page template that could potentially be used for all site pages other than the homepage. [Figure 3–15](#) shows a generic example of such a page template. The template is basically same as for the homepage ([Figure 3–14](#)), but the content area is now a single placeholder to show page content in full.

Figure 3–14 Generic Page Template for the Home Page of a Web Site



Generic page template often used for the homepage of a web site, with a banner image at the top, a navigation fragment on the left, a footer fragment at the bottom, and a number of small placeholders in the center, each of which representing some "teaser" content.

Figure 3–15 Generic Page Template for All Other Pages of a Web Site

Generic page template often used for all pages of a web site other than the homepage. The template is basically same as for the homepage as shown in [Figure 3–14](#), but the content area is now a single placeholder to show data in full.

3.13 Subtemplates

Subtemplates are the same as page templates, but with one important difference: subtemplates do not have <HTML>, <HEAD>, and <BODY> sections. As such, they are essentially chunks of HTML code that can be inserted in page templates. Subtemplates may contain very simple HTML code, but they can also be quite complex, with their own scripts and the like. The code in a subtemplate is treated exactly as it would be when placed directly in a page template.

As shown in the site object hierarchy ([Figure 3–4](#) on page 3-9), subtemplates can only be placed within placeholders, and subtemplates may contain their own placeholder(s). Subtemplates are typically used as a method of dividing a placeholder (that is, contribution region) on a page template into further smaller, reusable areas with their own placeholders, as shown in [Figure 3–16](#). A placeholder can contain a subtemplate that contains one of more other placeholders, each with their own subtemplate or region template.

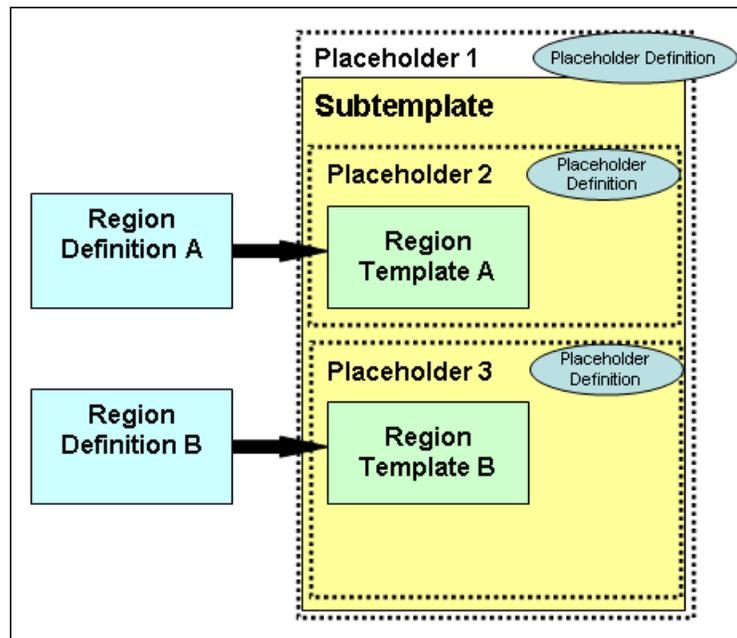
Figure 3–16 Subtemplate in a Placeholder

Diagram of a placeholder with a subtemplate that contains two placeholders of its own, each with a separate region template and region definition.

Please note that a good site design does not necessarily need subtemplates, and many web sites do not use them at all, since the addition of a subtemplate does mean that the designer has an additional type of site asset to manage.

Subtemplates can be used to help reduce the number of page templates used in a web site. This would be done by creating one main page template that can be used as broadly as possible for ease of reuse, and then in certain cases using subtemplates to change a placeholder on the main page template into multiple placeholders using subtemplates. This further allows reusability. A site designer can create a large area with one placeholder, which can then be used and reused with a placeholder containing a subtemplate with multiple placeholders in different layouts.

3.14 Index Pages and Flex Pages

Index Pages

An **index page** of a site section is its landing page; that is, the page that opens when a visitor first enters that section. Sections in the site hierarchy usually have an index page assigned to them, but this is not required. For example, a search results page, which you do not want users to browse to directly, may only have a flex page. You assign a page template as the index page of a site section.

The information on an index page is statically associated to the page. Contributors can change the contributor data file on the index page using the Contributor editor, and native documents using their associated third-party application.

Flex Pages

Flex pages are optional for site sections, and they are typically used to dynamically present content on a web site. A flex page can have static content, but what makes flex

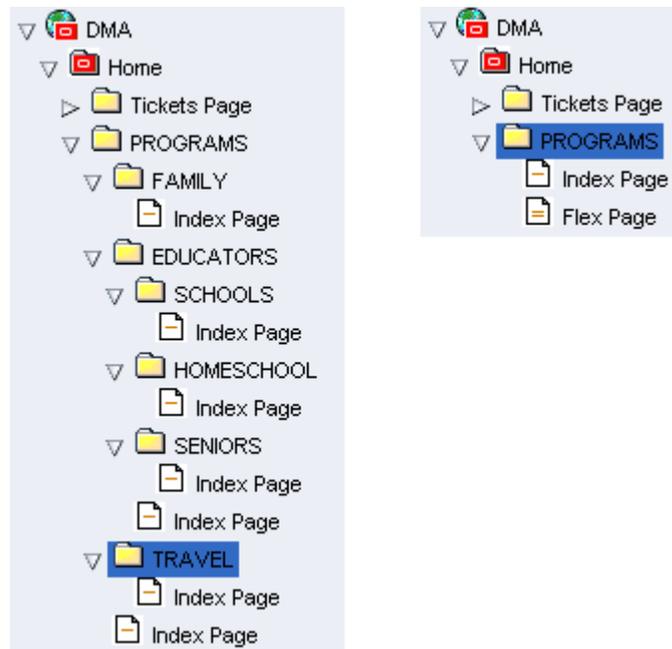
pages useful is their ability to have dynamically placed and replaceable content. They are used to create multiple versions of the pages within a site section; they provide a different content view for a site section. Flex pages allow you to handle large sites without needing to physically create thousands of pages. A comparative example can be seen in [Figure 3-17](#).

Flex pages are required if you allow contributors to add new contributor data files or native documents (both of which amount to new web pages) to the web site. These files are made available to the site when they are picked up by a dynamic list, a search, or the target of a link.

Since the best use of Oracle Site Studio is through reusing assets, it is very possible that you only have one page template across an entire site. The index page would have a region whose content is stated within the project file; the flex page content in the contribution region is specified by stating the content ID in the URL of the page. This means that the URLs of flex pages are not "reader friendly" because the content ID must be stated in the URL, unlike index pages, which are loaded when you simply use the URL of the section.

You can create a site comprised entirely of index pages, but then you must create sections with new index pages in order for the site to grow. By using index pages, your site can grow on its own from additional content submitted by contributors, and you, the designer, do not have to do anything. Your site becomes much more scalable with flex pages.

It is possible to get the same type of flow within your web site by using multiple index pages rather than a flex page. However, this type of setup would produce the same results at the cost of having a completely unmanageable site hierarchy. At best this would create more drag on the server performance in delivering pages, and more typically it would create confusion about where pages are located. Where previously something was located under <http://www.example.com/programs/>, it is now located under <http://www.example.com/programs/family/>, <http://www.example.com/programs/educators/schools/>, <http://www.example.com/programs/educators/homeschool/>, and so forth. This kind of site hierarchy would become unmanageable using only index pages. Flex pages allow you to control (in this example) all of your products under the simple <http://www.example.com/programs/> with only one index page and a flex page (compare hierarchies in [Figure 3-17, "Comparing Hierarchies Using Only Index pages and Using Flex Pages"](#)). The best use of Oracle Site Studio is in its ability to increase asset reuse and simplify the site structure to allow for ease of site management and maintenance.

Figure 3-17 Comparing Hierarchies Using Only Index pages and Using Flex Pages

3.15 Named Pages

A **named page** is a flex page that has been assigned a name, and this name is mapped to the URL containing the specific content ID that the same flex page would use. Thus rather than using a URL such as

```
http://www.example.com/products/ACD210049621040BDG
```

you can map a name to that page, such as 'widget'. Thus you could point people to

```
http://www.example.com/products/widget
```

and the mapping, stored in the project file ("[Project Files](#)" on page 3-28), would automatically redirect to the associated flex page URL. Each named page is specific to that section, and only to that section. This way it is possible to use the same name in multiple sections, but that same mapping will be to flex pages in the separate sections. It is not possible to have a named page in one section map to a flex page in a separate section.

It is important to note that when creating the named page mapping, that if you try to access the page through the normal method of referencing a flex page with the URL referring to the content ID, the browser will display the mapped name URL instead.

It is not possible to create a named page that maps to a flex page anywhere but the current section, including all subsections.

3.16 Contributor Data Files and Native Documents

The content of Oracle Site Studio web sites is stored in either of two file types: contributor data files or native documents. These content files are stored on the content server, and they are the files that contributors interact with when adding or editing web site content. You, as the site designer, decide which of these (or both) are used for a particular web site.

The content files for a web site (contributor data files or native documents) are initially selected (or created) by the designer or contributor, and modified by the contributor. Contributors may be able to switch the content file associated with a contribution region if the designer has specifically allowed this in the placeholder definition associated with that region.

When a web page displays site content in a contribution region, it displays the content through a region template and region definition, which define where and how each named part of a content file displays within the page. You can assign a unique contributor data file or native document to contribution regions, or assign the same file many times, depending on whether you want to reuse specific content on your site. You choose the name and metadata for the content file when you first create it and check it into the content server. The names that you use for content files may be useful when managing these files on your site. Also, the names that you choose may depend on how often you are reusing content. More information on how to best name site assets is in [Chapter 4, "Efficient Web Site Planning."](#)

Contributor Data Files

Contributor data files are XML files that are created by Oracle Site Studio. Each contributor data file is associated with one (and only one) region definition that defines its 'content class' in terms of its constituent elements (see [Section 3.10, "Region Templates and Region Definitions"](#)). For example, if a region definition comprises four elements called Title, Subtitle, Body_Text, and Image, then all contributor data files associated with that region definition contain these same four elements. Each element in the contributor data file can be edited in accordance with the element definition associated with that element in the region definition (see [Section 3.9, "Elements and Element Definitions"](#)). The element definitions specify the editing options that are available to contributors when they edit an element in a contributor data file. Some elements may be set up to be plain text only with very limited formatting capability, while others might be WYSIWYG (What You See Is What You Get), which generally offers a much broader editing experience.

The content of contributor data files can easily be reused on a web site; that is, it can be displayed in different locations on a site, either completely or partially (different elements), depending on where it is used. It is worth noting that when contributors edit a contributor data file, they see *all* its elements in the Contributor editor, even if only some elements are actually displayed in the contribution region being edited. The other elements may be used elsewhere on the web site, so editing that information may affect other pages on the site.

See *Using Oracle Site Studio Contributor* for more information on working with contributor data files.

Native Documents

Native documents are files that are created using familiar third-party applications such as Microsoft Word. They are converted to HTML format using Dynamic Converter so that they are viewable on the web site. Dynamic Converter uses conversion rules and templates to decide how to convert a native document. Native documents are edited using their associated application (for example, Microsoft Word for '.doc' files).

The content of native documents can also be reused on a web site, although native documents are generally not as flexible in their reusability as contributor data files. Contrary to contributor data files, native documents are not necessarily segmented into small, reusable chunks of information; their structure is often much more "free-flowing." That said, judicious use of styles and other formatting features in Microsoft Word may overcome some reusability drawbacks of native documents. One important benefit of using native documents is that most contributors are already

quite familiar with, say, Microsoft Word, which makes this application an easy and convenient editing environment for the site content.

For more information on native documents, see [Section 10.8, "Working With Native Documents."](#)

3.17 Fragments

Fragments are chunks of code that enhance the functionality of an Oracle Site Studio web site. They are essentially containers for HTML, Idoc Script, JavaScript, JSP, ASP, and referenced files (images, CSS, includes, and so on). Examples of fragments would be a breadcrumb trail, a navigation bar, or a copyright notice in a footer.

The fragment specification is written in XML and is stored in a fragment library (along with other fragments, if you choose). The fragment library comprises a single XML file that describes its contents and a zip file that stores all of the assets used by each fragment. Fragment libraries are stored in the content server. Several default fragment libraries with many predefined samples are automatically checked in when you install the Oracle Site Studio component on the content server.

The predefined fragments come in four categories: navigation fragments, dynamic list fragments, static list fragments, and other fragments. Each category contains a variety of fragments in multiple scripting languages. You can use any fragment as-is or copy and edit the fragment to suit your needs. You can also create fragments from scratch. For details on the fragments provided with Designer, see [Appendix C, "Sample Fragments."](#)

As you start building your own fragments, you should create your own fragment libraries for these fragments. This offers several benefits:

- You can easily track and organize where your fragments are.
- You can easily move, copy, or back up your fragments when they are stored in your own fragment library.
- You will not inadvertently alter a fragment created by another designer, which affects the web sites currently using that fragment.
- If you create a fragment from an existing fragment, you can easily revert to the original fragment, if need be.

For more information, see [Chapter 13, "Working With Fragments."](#)

3.18 Cascading Style Sheets

Cascading style sheets (CSS) are a popular method to control positioning and layout of a page template. CSS files can be used in Oracle Site Studio web sites. They are separate site assets that are stored and managed on the content server. CSS files can be edited directly in Oracle Site Studio Designer. When a CSS file is selected for editing, it opens in source view.

To use CSS files on your web site, you must reference them directly on a page template, or you can wrap them in a fragment and include that on a page template.

An alternative method of controlling page layout is to use tables to place objects in specific locations on a web page. The advantage of using tables rather than CSS is that tables allow for more specific placement of objects in a more straightforward manner. Table-based page templates look more "natural" in design view than CSS-based page templates. However, tables can become very complex to create certain layouts, and also do not allow for other types of control that a CSS file would. For example, CSS

enables you to control placement, fonts, paragraph styles, alignment, backgrounds, cell actions, and much more.

When CSS is used to control layout, the applicable CSS styles for the item that the contributor is editing are made available in the toolbar (if the designer has made that portion of the toolbar available in the element definition).

Specific information on the capabilities of CSS is available online at www.w3c.org.

3.19 Project Files

Project files are XML files that store all information about an Oracle Site Studio web site on the content server. In fact, when you connect to a web site in Oracle Site Studio Designer, you essentially connect to its project file on the content server. Project files should never be edited outside of Oracle Site Studio Designer or Manager!

Project files store lots of site-related information including:

- Site hierarchy
- Properties of each site section (including associated page templates, region templates, and content files, and custom section properties)
- Explicit data file associations (that is, what content files are used where in the site)
- Mappings of placeholder names to placeholder definitions
- Items in the asset pane (including how they are categorized)

Efficient Web Site Planning

Oracle Site Studio Designer 11gR1 uses site assets to make the maintenance and upkeep of web sites as efficient as possible. To best create a web site in Oracle Site Studio, some design points should be remembered when planning the web site and the site assets of the web site.

Planning is key to building a successful web site with Oracle Site Studio. Before you begin inserting text, graphics, and scripts into your page templates, you should ask: What is the function or role of the site? Is it a department-level site, a company-wide site, an internal site, an external site? How many users will visit the site? How many users will contribute to the site? Will there be different security access levels for each contributor? Do you plan to replicate or publish the site? Is the site expected to grow over time?

These are all important questions to ask before you begin. We cannot predict your particular needs, but we can suggest some key points that you should consider before developing your site:

- [Section 4.1, "Planning the Web Site"](#)
- [Section 4.2, "Naming Site Assets"](#)
- [Section 4.3, "Planning the Contribution Model"](#)
- [Section 4.4, "Order of Site Asset Creation"](#)
- [Section 4.5, "Creating Element Definitions"](#)
- [Section 4.6, "Creating Region Definitions"](#)
- [Section 4.7, "Creating Region Templates"](#)
- [Section 4.8, "Creating Subtemplates"](#)
- [Section 4.9, "Creating Placeholder Definitions"](#)
- [Section 4.10, "Creating Page Templates"](#)
- [Section 4.11, "Planning the Site Hierarchy"](#)

The information in this section is about the functionality introduced in Oracle Site Studio 10gR4, and does not apply to previous versions of Oracle Site Studio.

4.1 Planning the Web Site

When creating a static web site, just using pure HTML, little thought is typically required to planning the way the site flows. One of the major advantages that Oracle Site Studio offers the designer is reusability of the site assets. This does mean that the most involved, and most important, aspect of creating a web site is planning the site.

All of the aspects of the web site should be considered, including where certain information will display (even if the specific information is not yet determined).

Proper planning of the web site helps determine how to maximize reusability of the web site through appropriate use and reuse of page templates, subtemplates, and region templates.

You should ask yourself the following questions as you plan the web site:

- [Section 4.1.1, "Why Is Planning Important?"](#)
- [Section 4.1.2, "What Parts of the Site Will Be Reused?"](#)

4.1.1 Why Is Planning Important?

It is important to understand that the more time spent on the planning process, the easier the web site and the site assets are to create and manage. You might think that there is more time required to plan a managed web site using Oracle Site Studio, but the result is much less time spent managing the assets.

Proper planning is vital to making the site easier to run. In the beginning stages, it might seem as if more time is being spent before any pages are complete compared to older methods of creating a web site. But the results of time spent properly planning the web site makes the construction of it through the site assets much easier, and the maintenance of the web site, especially with making later changes when the web site is live.

4.1.2 What Parts of the Site Will Be Reused?

When you consider the web site, you should look at all content and all of the structure and consider what is reusable, and what should be used only once. When considering this, it could be thinking of simply the layout of the page, or it could be simply what data is displayed, or it could be a consideration of a certain piece of data displaying in a certain way.

Because there are so many ways of arranging and reusing the different parts of the site, it may be helpful to look at these examples of organization and reuse to think about while you consider your own web site.

In a typical web site, there is the navigation on the left, the banner graphic on top, and a large central area with the information on the page itself. We would expect that the banner and the navigation should be on all pages, so this would be placed on the page template itself. But the information in the middle will obviously be different from page to page. This is where the considerations are most important.

The way the information is organized is the most important consideration. When you look at one page, it may have objects arranged in one column, or in an array, or broken up with images. It's possible to arrange everything in one placeholder, but there is the other aspect, where you can create smaller sections, each with its own smaller contributor data file.

Consider a page on your web site that would list open employment positions. You could create the page such that it is one placeholder, listing all internal positions and all external positions. Or you could create a subtemplate within that placeholder (which would then contain separate placeholders and region templates, and so forth), so that the external positions would be stored separately from the internal positions. Each could be maintained in a separate contributor data file, so that the external web site would contain only the external announcements, and the internal web site would contain both the contributor data file with external announcements and the one with internal announcements.

Another use would be where each department in the company could list their own job openings; then, on one central page, you could collect all of those individual openings and display them all. In these instances, you can use a subtemplate to easily manage the differing numbers of placeholders.

Other considerations for how you lay the data out on the page, and how to organize the placement of the data within the web site, needs this kind of consideration on a page by page basis.

You should consider these questions: Would it be best to use one placeholder on the page template, then use a subtemplate to break that placeholder up into parts? Or would it be better to have a few more page templates to allow for different placeholder arrangements?

Another example would be an instance where you have a small piece of information that does not necessarily need a separate page, but you would definitely want to reuse. An example of this could be stockholder contact information, or possibly job application information, separate from typical corporate contact information. The information is not enough to necessarily warrant its own page.

In all of these cases, the page template would be the same. It would have the banner, the navigation, a footer, and then in the middle, the placeholder representing the area that can be replaced and filled with any information you need, structured exactly as you need it. It was the consideration of how to use a subtemplate to further use a placeholder or multiple placeholders within that template that enables you to keep the single look that you need for all pages.

It would also be possible to achieve this layout with different page templates on each page. Again, it depends on how you plan your site.

As you can see, the most important part of the site creation is to figure out how each portion of the web site, both in terms of structure and content, is displayed. With Oracle Site Studio, the more time you invest in planning before you create, the less time you spend creating the hundreds and even thousands of pages your web site delivers.

4.2 Naming Site Assets

Site assets in Oracle Site Studio Designer are intended to be reused. Since the asset itself can be used in not only different areas of a web site, but also in different web sites that you are designing, the naming of each asset is important. With a good naming scheme, assets can be more easily managed and more easily deployed in one or many areas of a web site, and one or many web sites.

The best way to name an asset is to describe what it does and how it functions, rather than by using a name based on where an asset is used in a web site.

The first thought of many designers is to put the name of the web site in the name of each asset. This would group them when listing the contents of the content server. But the difficulty here is when you run multiple web sites, since all of the assets are reusable. Rather than create an entirely new set of assets and definitions for a certain section in your web site, you can easily import the other assets you've already created. In doing so, though, when the assets have the name of the other web site in the name, this can cause some issues of continuity, understanding, and even some confusion regarding why some assets are present. Names of the assets should also avoid the place of the asset within the web site.

Figure 4–1 Hierarchy Showing a Poorly Named Template



In the image, you can see the issue in naming an asset based on location. By naming the page template originally intended to be the primary page template as "primary_template," we caused some problems when we realized that it would also be the template used on the secondary pages. Proper planning would have eliminated this issue, giving the template a name such as page_template_with_2_placeholders.

Consider these two different methods of naming the site assets in your project:

Naming convention with little planning toward maintenance:	Naming convention with considered planning toward maintenance
placeholder_def_1	element_def_WYSIWYG_fulledit
p_def_2	element_def_imageonly_minedit
default_definition	placeholderdef_subtemplate_newslis
front_page_template	regiondef_customform
subtemplate_up	regiontemplate_title_and_body
element_title	pagetemplate_home
region_replace	pagetemplate_errorhandler

Future updates are much easier to handle when the assets are named to describe what they do, allowing you to place them as needed. This also means you do not have to look in all managed web sites for an asset that you have already created.

If you are planning on sharing Oracle Site Studio assets between Oracle Site Studio Designer and the Oracle Site Studio extension for JDeveloper, you should be aware that some characters do not work in JSP pages. For this reason you should not use operators and words reserved by Java (for instance, '-' or '+'). For more information on the Java requirements, see the following links:

<http://java.sun.com/docs/books/tutorial/java/nutsandbolts/>

and

<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/JSPIntro7.html>

You should also be aware that you should avoid using multi-byte characters (MBC) in any value that may become a URL. Oracle Site Studio does not support MBC in fields where the value may become part of the URL. The *SiteID*, *ContentID*, *URLPageName* and *URLDocName* fields are examples of fields that become URLs. Fragments may use MBC internally, but should not be used if the fragment generates a URL.

Additionally, if you already have data files or native documents with MBC in the content ID then you will need to use a different method to reference the document, such as the `SSUrlFieldName` configuration option to use a metadata field other than the `dDocName`.

4.3 Planning the Contribution Model

As you plan the site hierarchy and prepare the assets for the web site, you must think about a "contribution model." That is, how content gets placed on the site by users (contributors and managers).

You should ask yourself the following questions as you plan the contribution model:

- [Section 4.3.1, "How Much of the Site Will be Provided by the User?"](#)
- [Section 4.3.2, "How Much Control Will the User Have?"](#)
- [Section 4.3.3, "Will There Be Different Security Access for Each User?"](#)
- [Section 4.3.4, "Will Contributors Submit Native Documents?"](#)
- [Section 4.3.5, "How Will the Contribution Process Be Coordinated?"](#)

4.3.1 How Much of the Site Will be Provided by the User?

There are some things that the designer must add to the site, such as the background color, positioning devices (HTML tables or CSS), site navigation, and custom scripts. But much of the content for each web page (that is, the actual information on the page) can be created and edited by a contributor.

To really harness the power of Oracle Site Studio, you should open up as much of the web site as possible to these users. This way, your web site can be continuously updated without the bottlenecks or delays typically associated with a web site.

4.3.2 How Much Control Will the User Have?

On every page template, you can have one large contribution region or several small contribution regions. Within a region template, you can have one or several elements, each one appearing as a field in Contributor where users (contributors) add and edit content.

There are different types of elements that can be used for specific purposes, like adding and editing text, graphics, and updating lists. You can turn on or off certain formatting attributes, such as the choice of typeface, font size, images, tables, and custom properties. These choices depend on how much control you want the contributor to have over a given web page. You can also enforce what content is added using the validation feature or your own validation scripts.

In addition to these, you can allow users to modify the site hierarchy by enabling the Manager application on one or several pages.

You must balance this control with the desire for consistency, as dictated by your organization's guidelines.

4.3.3 Will There Be Different Security Access for Each User?

You should know your users, their role in the organization, and their knowledge of web publishing before you give them access to the site. You may want to provide equal access to all contribution regions on the web site for all users, or you may want to limit access for some and grant full access to others.

For example, you may want to provide full access to the entire web page for members of the marketing department and access to only a portion of that page to all other departments. To do this, you would set up different users on the content server, assign a different contributor data file or native document to each region, and assign unique security metadata (using the metadata framework on the content server) to those files.

As a result, only the files that a particular contributor has permission to edit will display a contribution graphic (Figure 4–2) on them on the web page when in contribution mode.

Figure 4–2 Contribution Graphic



Sections of the web site can also be marked so that only certain users can view them. This is handled in the section settings within the web site hierarchy. After the section has limited access, the section is available for viewing (and editing, if you want) to only those users that you want to have access.

4.3.4 Will Contributors Submit Native Documents?

In addition to using Contributor to add and edit content stored in Oracle Site Studio data files, contributors can also add native documents (created from applications like Microsoft Word, Excel, or PowerPoint) to the site. These documents are converted into web-viewable renditions using Dynamic Converter.

Contributors can add new and existing native documents directly to the web site (for example, by assigning one to a region, using the link wizard, or adding one to a dynamic list). They can also use existing methods in the content server to check in the document (the content server's content check-in, Desktop Integration Suite client, Oracle Content Server's folders functionality, and so forth).

Contributors can also edit the native documents directly from the web site (using the "Check Out and Open" option on the **Menu** icon of the contribution graphic and the edit document features in Contributor).

Native documents are displayed on a web page, and not in reusable chunks as the case with a contributor data file.

If you allow contributors to submit native documents to the web site (and again, you control this in a contribution region, an element, and a list fragment), you should educate them on how they should use the native document and any style guidelines you may be enforcing.

4.3.5 How Will the Contribution Process Be Coordinated?

If you are the designer, manager, and contributor of your web site, you know when and where the site must be updated. If, as the designer, you are working with a single contributor or manager, you must communicate the required updates for the site. Specifically, you should inform them of certain changes to the site. The contributor, however, may also alert you when content has been added.

If you are working with multiple users, communication can quickly become complex. You must include some kind of communication process between users. One solution is to introduce a formal review process using workflows. This means that various persons review and approve site changes before they become "live."

If you would like to give the contributors a brief overview of how Oracle Site Studio works, you may want to distribute *Using Oracle Site Studio Contributor* to them.

4.4 Order of Site Asset Creation

After the web site has been planned, the assets used to construct the web site in Designer should be created. For the most efficient use, it is recommended that you create these assets in a particular order.

The most efficient method of asset creation for most web sites is to build the assets "from the bottom up:"

1. Element definitions (see [Section 4.5, "Creating Element Definitions"](#))
2. Region definitions (see [Section 4.6, "Creating Region Definitions"](#))
3. Region templates (see [Section 4.7, "Creating Region Templates"](#))
4. Subtemplates (see [Section 4.8, "Creating Subtemplates"](#))
5. Placeholder definitions (see [Section 4.9, "Creating Placeholder Definitions"](#))
6. Page templates (see [Section 4.10, "Creating Page Templates"](#))
7. Site hierarchy (see [Section 4.11, "Planning the Site Hierarchy"](#))

Of course it is not entirely necessary to create the assets in this order. However, you typically create the web site more quickly and efficiently if you follow this order.

4.5 Creating Element Definitions

The most efficient assets to define first are the elements. Elements are the assets that define the editing interface that the contributor uses. The specific use of each one is handled through element definitions. By creating an element definition, you are defining how the contributors work with the data files by defining the editing regions and the toolbars accessible to the contributors.

In the element definition you can control exactly which editing features are available to the contributor in the toolbar. For instance, you might not want the contributor to be able to change the font size or color, but still be able to use bold and italics. Generally you define at least one of each kind of element (WYSIWYG, text only, image only, static list, dynamic list, and custom) and usually, you define two or more of each type of element to allow for the different toolbars in Contributor.

The element definition is used to control the toolbar that the contributors can access to edit the pages. However, it might be that you want to have several different definitions for each element so that you can have multiple options of toolbars for the contributors, rather than one or two. It is important to consider the role of the contributor and how much they contribute on different pages.

It is highly recommended that you name the element definitions carefully, as naming the elements with respect to place on the page or with the name of the site in them can make reusing them less intuitive. It is not recommended that you name the element in terms of its place on the page, within the web site, or even by using the name in the web site within the element definition. The optimal method for naming an element definition is to list a relative amount of access in the toolbar, such as "element_WYSIWYG_fulledit" or "element_text_undo_only."

Notice that the element definitions were named starting with the word "element." That is to make all elements appear together when searching for them in the content server. Using other naming conventions such as including the name of the web site would

also group the assets in the web server, but this would make the definitions less intuitive to manage if you use the same definitions for multiple web sites.

4.6 Creating Region Definitions

You should create region definitions before you create region templates. A definition (region definition, placeholder definition, element definition) controls which data is displayed and how the contributor interacts with the data. The template (region template, subtemplate) control how the data is displayed. The region definition, specifically, controls which elements (WYSIWYG, Image only, and so forth) open in Contributor. The region template is used to arrange the placement of the elements and how that particular data is displayed on the web page.

As you create a region template, you are asked to associate it with a region definition. If the definition is not already defined, the region template is not easily associated with the definition afterward.

The typical region definition contains multiple elements, and various combinations of these elements are used on several different region templates. That is, the region definitions are as broad and encompassing as possible. This allows multiple region templates to access the same grouping of elements (but it does not have to be the same data file) and arranging them in different ways. If you add many element definitions to a region definition, however, you should be aware that all element definitions listed in the region definition open in Contributor, even if only a few are visible on the page.

Because the region definitions limit the availability of elements in the contribution model, it is especially important to follow good naming conventions for region definitions. This helps you sort out which definitions are best suited for the different region templates you create.

4.7 Creating Region Templates

Region templates are the assets that format the data files into HTML. You use region templates to place the content, such as text, images, or other simple items, that would not be inside of an element, and the elements. The region templates are used to arrange the elements (limited by the region definition) into the layout desired to be reused as you see fit in the different places in your web site.

The region template has available to it only those elements that have element definitions listed in the region definition for the template. However, the region template does not have to use all elements listed in the definition. This allows for a great deal of variety in the different templates available.

It is likely that region templates are the largest number of assets you create for the web site. Because of this, the naming convention you use in creating the region templates is important. The name should help define what the template does, not where on the page it is used or where in the web site. A name such as "regtem_titlebodyview" or "rt_bodynoimage" would likely be a more useful name when maintaining the site than would, say "regiontemplate14" or "RT_SupportForum."

Since region templates can contain HTML and images and other assets (for example, a region template-specific CSS) that are not part of the data file that is passed through the elements, the templates can be used to incorporate some information and text that is not editable by the contributor. However, it is important to understand that any of that information is not part of the contributor data file that the contributor edits. This means that as you reuse a region template, the data associated with it opens in the same manner regardless of what data is associated with that region template. It also means that the designer is the only person who would be able to modify that

particular data. The contributor would be able to create and edit any data that is assigned to the elements in the region template through the placeholder, but anything that is placed outside of the elements, but still within the region template, would be static.

4.8 Creating Subtemplates

Subtemplates are major portions of HTML that can contain placeholders. This makes them very useful as a way to reduce several page templates while still maintaining several different ways the pages in the web site can appear.

Subtemplates can only appear in a placeholder. Subtemplates can only contain a placeholder, or static content such as HTML or an image. A subtemplate can also reference a CSS file.

You should ask yourself the following questions as you plan subtemplates:

- [Section 4.8.1, "Is a Subtemplate Necessary for the Site?"](#)

4.8.1 Is a Subtemplate Necessary for the Site?

One of the benefits of using a subtemplate is that you can reduce the number of page templates used in a web site. Through a subtemplate, you can have the same page template used for both primary and secondary pages.

This is one reason why planning is essential. While a subtemplate can reduce the number of page templates, it also means that you have a subtemplate to manage for the web site. Fewer page templates means that site-wide changes are easier to make. Subtemplates can also add a level of static information because you can add HTML, images, and other similar static site assets to a page that you may want placed on a certain number of pages (such as the secondary pages) that would otherwise have to be managed separately from the subtemplate.

The subtemplate can also help you place multiple placeholders within a single placeholder. Since a placeholder can contain a subtemplate, and a subtemplate can contain a placeholders, then it could be that you want the subtemplate present for future web site expansion that is still to be determined.

4.9 Creating Placeholder Definitions

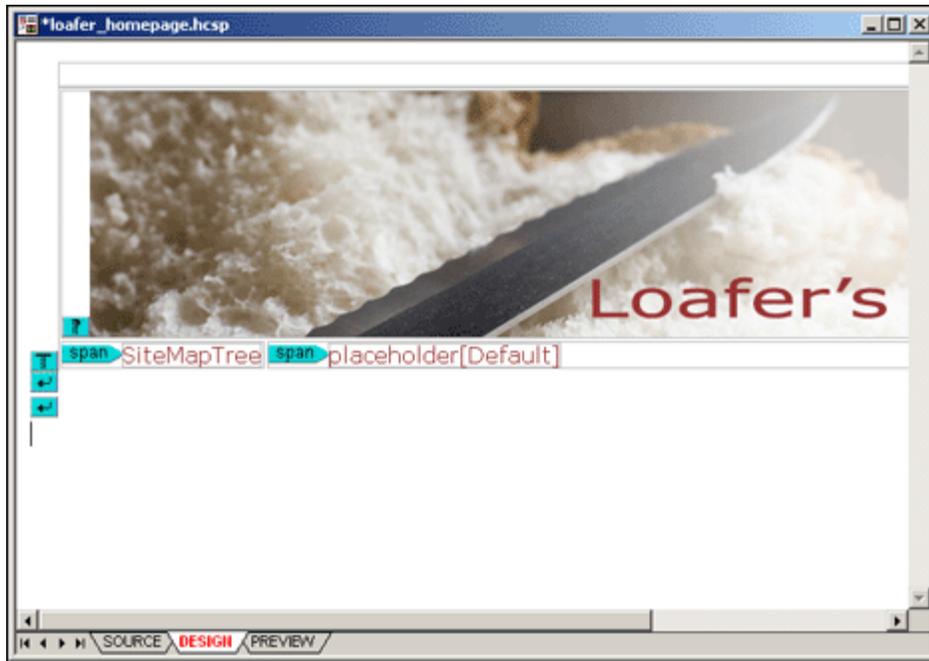
Placeholder definitions are what connect each placeholder to the content and other assets associated with that placeholder to display.

You should ask yourself the following questions as you plan placeholder definitions:

- [Section 4.9.1, "How Will The Placeholder Function On The Page?"](#)
- [Section 4.9.2, "What Will The Placeholder Definition Control?"](#)

4.9.1 How Will The Placeholder Function On The Page?

The placeholder, not to be confused with the placeholder definition, is simply a mark on the page in the Designer application. Placeholders are just conceptual boundaries in Designer.

Figure 4–3 Placeholder as seen in Designer

As you see in [Figure 4–3](#), the placeholder is just a marker, using the word "placeholder" and also lists the name of the placeholder. The actual amount of space that a placeholder uses on a web page is determined by the size of the content placed in it.

The data associated with the placeholder depends on which page the placeholder is on, which happens after the site structure is complete and you begin to place content in the sections of the web site.

It is the placeholder definition that determines how the placeholder functions, and what the placeholder contains. This means that you should view the placeholder as an available section for you to use and reuse content in broad areas. It is common to have a page template, for instance, with only the navigation and the header image and nothing but a placeholder in between.

In [Figure 4–3](#), there is only a banner, a side navigation piece that is a fragment, and the placeholder. All of it is arranged using a table. This basic design could reliably be used as the entire structure of a web site, depending on how you use the placeholder.

This placeholder would then be easily replaced on any number of pages with the content specific to those pages. The content can be broken down into smaller placeholders if your design planning has led you to create a site based on a minimal number of page templates and using multiple placeholders and subtemplates in different ways.

4.9.2 What Will The Placeholder Definition Control?

The placeholder is little more than a tag that is meant to define an area on the page where the content is shown. The placeholder definition lists which region definitions (and their associated region templates) and which subtemplates are available to pass information through.

You can also control other details on a placeholder-by-placeholder basis through the placeholder definition. Some of these items include workflow, whether the contributor

can edit the data displayed through the placeholder, whether the contributor can edit the metadata and the ability to view web site usage reports.

4.10 Creating Page Templates

A large number of web sites can be reduced to two simple page templates: the home page, and then all other pages. It is not uncommon for a web site to have a home page that is distinct in terms of layout and design compared to the other pages in the web site.

It may be the case that there are multiple templates for your site. It is also possible to make an entire web site work with one page template. If you do design the web site with several page templates, you should possibly consider the use of subtemplates to reduce the number of page templates. Fewer page templates on a web site makes site-wide changes much easier.

You should ask yourself the following questions as you plan page templates:

- [Section 4.10.1, "How Should Fragments Be Used?"](#)
- [Section 4.10.2, "Will Primary and Secondary Pages Require Different Templates?"](#)

4.10.1 How Should Fragments Be Used?

Fragments are especially useful for maintaining dynamic content or content built with a custom script. Fragments could also be used for content that you would like to manage separately from the site assets and the contributor data files. This might be as simple as a copyright statement or as complex as a JavaScript menu. The number of fragments and the complexity of those fragments varies, depending on your site.

For more information on fragments, see [Chapter 13, "Working With Fragments."](#)

4.10.2 Will Primary and Secondary Pages Require Different Templates?

Primary and secondary pages can both use the same page templates. However, since secondary pages are the only pages that can have dynamically placed content, you should consider the effect on how you view your page templates (and even your placeholders and placeholder definitions) with respect to the advantages of dynamically placed content.

A secondary page serves as the backdrop for content added to the site by a contributor. Secondary pages are required if you allow contributors to add contributor data files or native documents (both of which amount to new web pages) to the web site. These files are made available to the site when they are picked up by a dynamic list, a search, or the target of a link.

It may be that you first build your site with just primary pages, saving secondary pages until after you set up contribution regions on the primary pages and know exactly what type of content contributors submit to the site. Then, you could add the secondary pages to handle this content.

Regardless of whether you use the same or different page templates for the primary and secondary pages in your web site, it is important that you name the page templates appropriately. This is the same for all other site assets in Oracle Site Studio.

Consider the example shown in [Figure 4-4](#):

Figure 4–4 Hierarchy Showing a Poorly Named Template



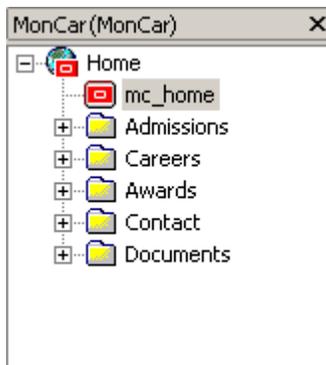
The same page template was used for both the primary and secondary pages. The page template name was based on where the template was based on initial placement, and when the site was expanded the reuse of the page template created a confusing arrangement.

The most efficient naming of site assets, including page templates, should be based on how the page template is used. Naming conventions based on where the asset is used in a web site (for instance, "page_template_primarypage") or based on in the order of creation (for instance "pagetemplate3") can make the assets harder to manage.

4.11 Planning the Site Hierarchy

The site hierarchy is the framework of your web site. You should give yourself plenty of time to plan the hierarchy before you start creating web pages. The site hierarchy not only helps you organize and manage content on your site, but it is also used by Oracle Site Studio to automate certain tasks.

Figure 4–5 Site Hierarchy Pane



You should ask yourself the following questions as you plan the site hierarchy:

- [Section 4.11.1, "How Deep Should the Hierarchy Be?"](#)
- [Section 4.11.2, "How Will The User Navigate the Site?"](#)
- [Section 4.11.3, "How Should the Sections Be Named?"](#)
- [Section 4.11.5, "Should There Be Both Primary and Secondary Pages?"](#)
- [Section 4.11.4, "How Reusable Should The Page Templates Be?"](#)

- [Section 4.11.6, "How Will Content Be Reused?"](#)
- [Section 4.11.7, "Will A Manager Be Necessary?"](#)

4.11.1 How Deep Should the Hierarchy Be?

A deep hierarchy places information at multiple levels where information is heavily categorized. This works well for large organizations or any organization that anticipates growth on the web site.

When planning the web site, a lot of care should be taken in considering the hierarchy of the web pages within the site. The web site hierarchy can be as deep as you want, and as deep as you want. You can create as many different sections from the home page as needed, and a section can contain as many sections as needed as well. However, the thing that should be kept in mind while designing the hierarchy is that while the structure can be as wide as needed (that is, sections can be nested to any depth), that this can create unwieldy URLs. The wider a section is nested in a web site, then the longer the URL is to retrieve that information. Usually this is a trivial consideration, but for some designers this can be a major point.

Each section listed in the site hierarchy can have a primary page, and a secondary page. Since secondary pages are the pages that have replaceable content, the secondary pages are used to create multiple versions of the pages within a section. The primary page within a section is the page that opens for that section, it could be considered the landing page for that section.

4.11.2 How Will The User Navigate the Site?

While you use your site hierarchy to manage your site, visitors use your hierarchy to browse to and locate content. In Oracle Site Studio, when you add a navigation fragment to a page template, the fragment reads your site hierarchy and generates links that comprise the overall site navigation. You can easily add, remove, and rename sections of your site, and these changes are seen in your site navigation. You should think carefully about the visitor's experience as you construct this hierarchy.

4.11.3 How Should the Sections Be Named?

Your site hierarchy has individual sections with names such as Products, Services, and About Us. These names are important. They not only help you organize content on your site, but they display in the navigation on the site, where visitors see it. It is a good idea to revisit these names regularly to ensure that they reflect the content of the sections they represent.

Another thing to consider as you assemble your site hierarchy is the web site address that contributors and site visitors see in their web browser address bar. By default, Oracle Site Studio uses the names (labels) that you give to the sections in your site hierarchy.

You can override these values, if you like, by specifying a different path name and page name for each section (see [Section 7.2.2, "Changing the Path Used in a Site Address"](#)). It is best to plan this ahead of time to minimize any late changes to the site address or paths used in the address, which could result in broken links or missing shortcuts for contributors and site visitors.

4.11.4 How Reusable Should The Page Templates Be?

As you plan your site, you can create new page templates for each section, or you can reuse page templates that are already being used in other sections. When you design to

use a small number of page templates, you then have fewer assets to manage, which makes the entire web site easier to manage.

By reusing page templates, you can maintain the look and feel of your web site in one or just a handful of page templates, and the changes are seen immediately across the site.

By creating new and unique page templates in each section, you might have more flexibility in the design of the site layout as a whole. You can create a unique design or slightly modify a different page template to accommodate the content that display there. However, if you make significant global changes to the site, then you have to modify multiple page templates. This might become a repetitive task that lends itself to mistakes and inconsistency.

If you choose multiple page templates, you should try to reuse fragments with referenced snippets as much as possible so that you can make global changes to just one or a handful of fragments.

However, through thoughtful use of subtemplates with the placeholders, it is typically possible to reduce the number of page templates.

4.11.5 Should There Be Both Primary and Secondary Pages?

There are several ways to handle page templates. You can assign one primary page and one secondary page for every section of your site hierarchy.

The primary page is the web page that users see when they go to that section on the web site (similar to the default page on a conventional web site). The information on the page is statically linked. Contributors can change the contributor data file or the native document on the primary page, or the information in the data file through the contributor interface.

The secondary page is a page that can have its content dynamically placed. A secondary page can have static content, but what makes secondary pages useful is their ability to have dynamically-placed content.

You can create a site comprised entirely of primary pages, but then you'll have to create new sections with new primary pages in order for the site to grow. By using secondary pages, your site can grow on its own from additional content submitted by contributors, and you, the designer, do not have to do anything. Your site becomes much more scalable with secondary pages.

Primary and secondary pages can have the same page template. The page template is not related to whether the page using the template is primary or secondary.

4.11.6 How Will Content Be Reused?

You can share and reuse contributor data files and native documents across a single web site and even multiple web sites in the content server. When you add these items to your site, you specify a specific location, or multiple locations by reusing associated site assets, where you would like them to appear.

You can even add content to your site that is not currently associated with any web site in the content server. You can show the same document on different sites, each with its own look and feel, without having to create multiple versions of that same document.

If you choose to do this, however, you must ensure that the content is appropriate for the web site and for public viewing. Workflows are good to implement to ensure this. You also must ensure that you have a secondary page with a replaceable region template in each section where the content appears.

4.11.7 Will A Manager Be Necessary?

You can create and manage the site hierarchy yourself, or you can hand this responsibility over to the site manager. Using the Manager application, site managers can create sections, modify sections, and more.

These changes can significantly change the appearance and behavior of the site. You should think about these changes and how they affect the structure of the site, workflow, replication, the role of contributors, and much more.

For more information on setting up Manager, see [Chapter 16, "Setting Up Manager."](#)

Getting Started With Designer

As the designer of the site, you spend most of your time using the Oracle Site Studio Designer application. It is recommended that you install Oracle Site Studio Designer locally on your computer.

In Designer, you build the site hierarchy and create the overall look-and-feel of the site. You make decisions about page layout, site navigation, and the use of fragments and contribution regions (that is, the editable areas of web pages).

This section covers the following topics:

- [Section 5.1, "Installing Designer"](#)
- [Section 5.2, "Starting Designer"](#)
- [Section 5.3, "Main Designer Window"](#)
- [Section 5.4, "Site Hierarchy"](#)
- [Section 5.5, "Properties Pane"](#)
- [Section 5.6, "Workspace"](#)
- [Section 5.7, "Site Assets Pane"](#)
- [Section 5.8, "Toolbox"](#)
- [Section 5.9, "Menu Bar"](#)
- [Section 5.10, "Toolbars"](#)
- [Section 5.11, "Log File"](#)
- [Section 5.12, "Site Connection Manager"](#)
- [Section 5.13, "Link Wizard"](#)
- [Section 5.14, "Switch Content Wizard"](#)
- [Section 5.15, "Fragment Editor"](#)
- [Section 5.16, "Contributor Data Files and Native Documents"](#)
- [Section 5.17, "Oracle Site Studio Manager"](#)
- [Section 5.18, "Oracle Site Studio Contributor"](#)
- [Section 5.19, "Oracle Site Studio Administration Page in Oracle Content Server"](#)
- [Section 5.20, "Oracle Site Studio Publisher"](#)

5.1 Installing Designer

For specific information on the system requirements for installing and running Designer, see [Section 1.8, "System Requirements."](#)

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

Note: You need administrative rights to the computer to install Oracle Site Studio Designer. If you want to install Oracle 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. Contact your system administrator for instructions on where to obtain the client software installer for Oracle Site Studio Designer.

If you have access to an Oracle Content Server 11gR1 instance, you can find the Oracle Site Studio Designer installation on the My Downloads page in the Oracle Content Server web interface (under My Content).

2. After obtaining the client software, run the installer executable.

The installation wizard is launched.

3. Follow the instructions on screen to install the Oracle Site Studio Designer software.

Note: If you install Oracle Site Studio Designer 11gR1 on a computer that already has an earlier Designer version on it, the Designer 11gR1 application installs alongside the earlier version. (The previous version is not removed.)

5.2 Starting Designer

To start Oracle Site Studio Designer, open the Start menu, select **Programs**, then **Oracle Universal Content Management**, then select **Oracle Site Studio 11gR1**, and then **Oracle Site Studio Designer**.

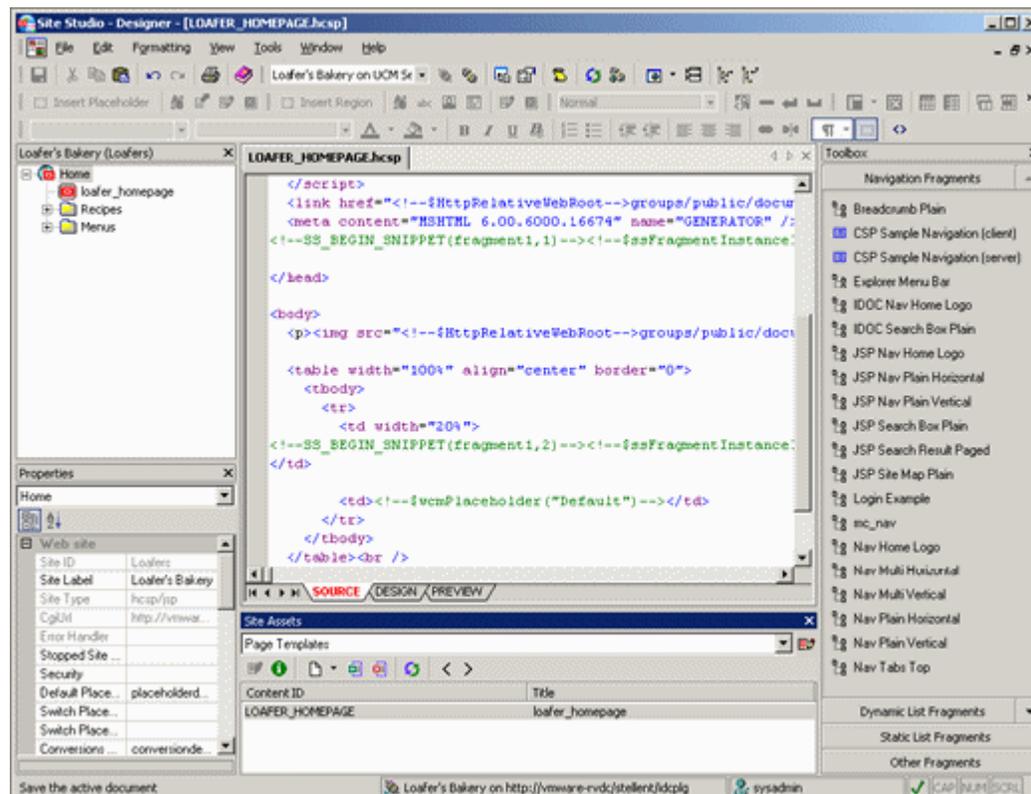
The first time the Designer application is launched, it opens an empty site hierarchy and workspace, and it prompts you to create a web site. Thereafter, Designer opens the last web site you worked on. You can prevent Designer from automatically reconnecting to the last site on the Customize dialog (Miscellaneous tab), which is called up by opening the **Tools** menu and then choosing **Customize**.

Note: Oracle Site Studio 11gR1 is fully backward compatible. This means that you can use Oracle Site Studio Designer 11gR1 to work with sites created in earlier Oracle Site Studio releases. It is important to note, though, that these sites continue to work in "legacy" mode; that is, they use the pre-10gR4 architecture and they do not take advantage of the architecture and features introduced in Oracle Site Studio 10gR4.

5.3 Main Designer Window

The Designer interface provides an environment where you can set up and manage all aspects of your site, including the site structure, page layout, navigation, and contribution. When you first open Designer, you see the site hierarchy and asset properties on the left, the workspace in the center, the site assets below that, and the toolbox on the right.

Figure 5–1 Main Oracle Site Studio Designer Window



Screenshot of the Oracle Site Studio Designer application. It shows the main interface, with menus and toolbars at the top, the fragments toolbox on the right, the site hierarchy and asset properties on the left, and the workspace and site assets pane in the middle.

The main Designer window consists of various elements:

- The **site hierarchy** is used to organize the structure of the site. This includes the home page, the individual sections, and the page templates associated with each section. For more information, see [Section 5.4, "Site Hierarchy."](#)
- The **properties** pane is used to view and edit a variety of settings for your site, including information about the site hierarchy, site assets, individual HTML and scripting tags, and so on. For more information, see [Section 5.5, "Properties Pane."](#)
- The **workspace** is used to view and edit site assets such as page templates, region templates, and element definitions. For more information, see [Section 5.6, "Workspace."](#)

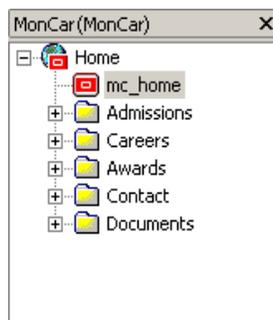
- The **site assets** pane is used to organize files that belong to your site (page templates, data files, CSS files, images, scripts, and so forth). You can use this pane to quickly and easily open, edit, and organize these files. For more information, see [Section 5.7, "Site Assets Pane."](#)
- The **toolbox** pane is used to create, edit, and manage fragments on your site. For more information, see [Section 5.8, "Toolbox,"](#) and [Chapter 13, "Working With Fragments."](#)
- The **toolbars** provide easy access to many features in the Designer application. For more information, see [Section 5.10, "Toolbars."](#)

The display and arrangement of these objects is a default. You can rearrange them and customize the interface if you like. For more information, see [Chapter 6, "Customizing Designer."](#)

5.4 Site Hierarchy

The site hierarchy shows the fundamental structure of your web site, and it ultimately governs many other aspects of your site (such as how you organize content and how Oracle Site Studio controls its presentation). You can show or hide the site hierarchy from the **View** menu.

Figure 5–2 Site Hierarchy Pane



Screenshot of the site hierarchy in Designer, which shows the structure of the web site in a Windows Explorer-like tree of site sections with their assigned primary pages and secondary pages.

After you specify a name and location for a web site in Site Connection Manager (see [Section 5.12, "Site Connection Manager"](#)), you are ready to start building the site structure. You do this by creating a home page, then individual sections (like Products, Services, and About Us) in the site hierarchy.

After you add the sections, you are ready to add page templates to each section. Page templates come in two flavors:

- **Primary page:** A primary page is the page template that is associated with a section in the site hierarchy. It is the page that you see when you go to that section of the site (much like the default page on a conventional site).
- **Secondary page:** A secondary page can also be associated with a section in the site hierarchy. However, its main purpose is to serve as a backdrop for files (contributor data files and native documents) added to the site by a contributor.

Whether you are adding a primary or secondary page to a section, you have three options: create a page template from scratch, reuse an existing page template, or create a page template based on an existing page template. For more information, see [Section 12.1, "Working With the Site Hierarchy."](#) You can also use the Manager application to modify the site hierarchy and assign page templates (see [Section 5.17, "Oracle Site Studio Manager."](#)).

5.5 Properties Pane

The properties pane displays useful information about the currently selected object in the site hierarchy, workspace, site assets pane, or toolbox. You can show or hide the properties pane from the **View** menu.

The information displayed in the properties pane is divided into categories, depending on the current object. [Figure 5-3](#) shows the property categories for a node in the site hierarchy called Careers. They are web site, Section, and Custom section properties. Other selected objects have different property categories. You can use the plus and minus symbols to expand or collapse the property categories to show or hide the properties and their values.

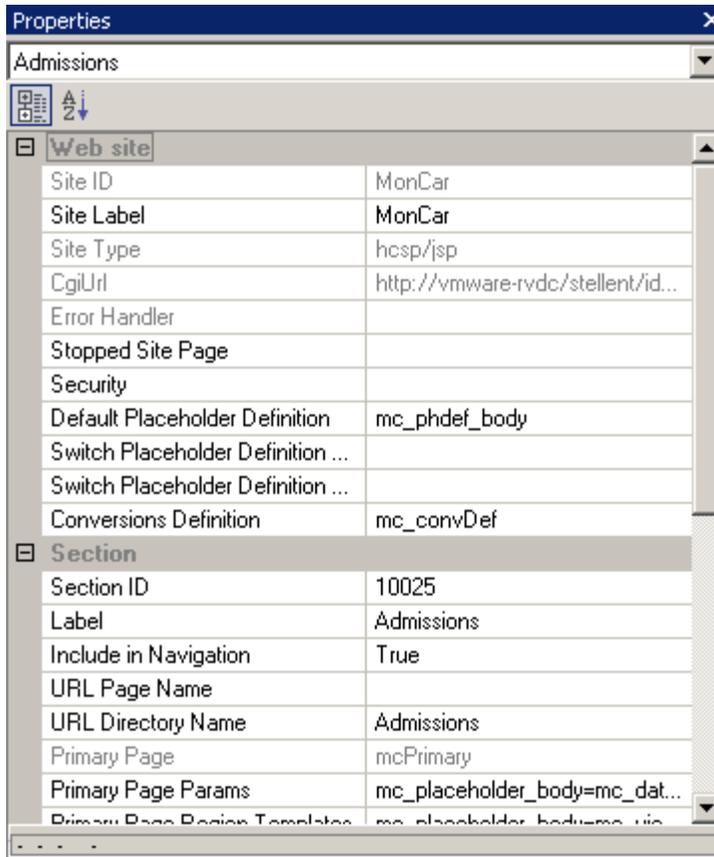
Figure 5-3 Property Categories



Screenshot of the properties pane for a node in the site hierarchy called "Careers," with three collapsed property categories: web site, Section, and Custom section properties.

The properties shown differ for each property category. For example, for the site hierarchy, it displays the site ID and label, Cgi URL, home page, default placeholder definition, and so on ([Figure 5-4](#)). For page templates, it displays information about the site assets, HTML tags, scripting tags, and fragments on the page. The site asset properties are shown as you work in design view so that you can see what is happening behind the scenes and make precise edits to specific tags.

Figure 5–4 Properties and Their Values



Screenshot of the properties pane in Designer, which shows the properties of objects in the web site in a grid, with the property names on the left and their values on the right.

You can use the properties pane to view and edit various settings on your web site, including the site hierarchy, primary pages and secondary pages, and the contents of site templates (HTML, scripts, fragments, and so forth) while working in design view. The properties pane is extremely useful to edit the contents of site templates (page templates, subtemplates, and region templates), especially when those changes cannot be performed directly in design view. Examples of this include background color, page margins, and table width.

You can use the icons at the top of the properties pane to show all properties in alphabetical order (by name) or categorized into groups of related properties. The values of some properties may be edited. These are shown in black. If a property cannot be edited, it is grayed out.

If you click a different area of a page template, region template, or subtemplate (while in design view), or a separate site asset, a different tag with different attributes displays instead. You can view the attributes of a parent tag by selecting it from the drop-down list at the top of the properties pane.

You can also use the Manager application to modify properties that relate to the site hierarchy. For more information, see [Section 5.17, "Oracle Site Studio Manager."](#)

5.6 Workspace

The workspace is where you generally do the most work. This is where you edit site templates (page templates, region templates, and subtemplates) and also define and configure other site assets such as region definitions, element definitions, and cascading style sheets.

When you open a site asset in the workspace, its associated file on the content server is checked out. After you are done editing and you save the changes (by clicking the **Save** icon on the toolbar or pressing Ctrl+S), the file is checked back into the content server.

You cannot open an asset to edit when it is currently opened by another user to edit. In that case, you will be alerted, and the asset will be opened in read-only mode.

The workspace in Designer provides the following features:

- [Section 5.6.1, "Dialogs vs. Tabs"](#)
- [Section 5.6.2, "Views"](#)
- [Section 5.6.3, "Source View"](#)
- [Section 5.6.4, "Design View"](#)
- [Section 5.6.5, "Preview"](#)
- [Section 5.6.6, "Form View"](#)
- [Section 5.6.7, "Right-Click Menus"](#)

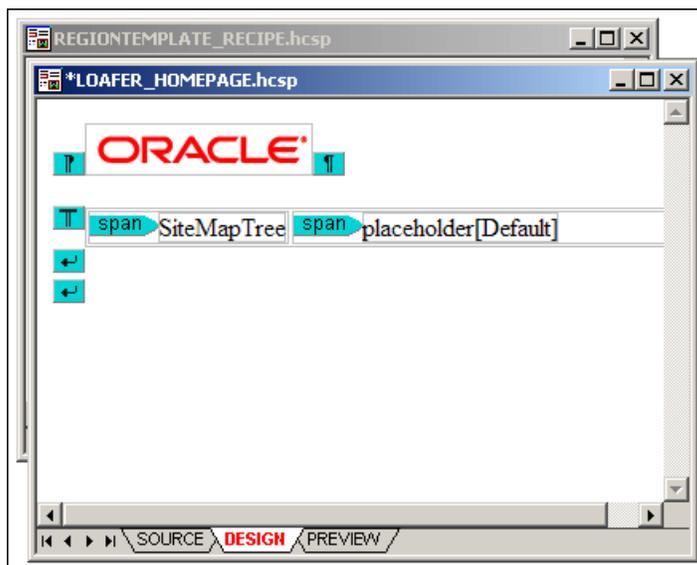
5.6.1 Dialogs vs. Tabs

The workspace can display site assets in two ways: as dialogs or as tabs.

Asset Dialogs

Site assets can display in the workspace as familiar Windows dialogs, with a title bar, minimize and maximize icons, and a close icon ([Figure 5-5](#)).

Figure 5-5 Two Assets Displaying As Dialogs



Two assets displaying as cascading dialogs in the workspace, with the top window having unsaved changes, which is indicated by an asterisk next to its name in the window title. Each window has minimize, maximize, and close icons.

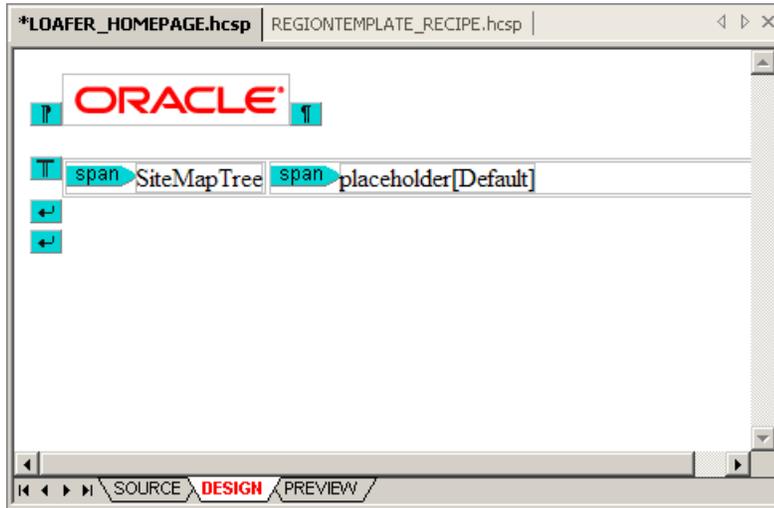
You can freely drag and drop dialog windows to any position within the workspace. This may be especially useful if you have a large screen, with lots of "real estate" to work with, so you can arrange the open windows to maximize viewing efficiency. You can also use the **Window** menu in Designer to change how the windows are arranged in the workspace (cascading, tiled).

If an asset has any unsaved changes, this is indicated by an asterisk (*) next to the asset name in the window title bar. When you close a window that has any unsaved changes, you are first prompted to save or discard these changes.

Asset Tabs

Site assets may also display in the workspace as tabs covering the entire workspace area (Figure 5–6).

Figure 5–6 Two Assets Displaying As Tabs



Two assets displaying as side-by-side tabs in the workspace, with the currently active tab having unsaved changes, which is indicated by an asterisk next to its name in the tab. There are left and right arrows in the top-right corner to cycle between asset tabs and an 'X' icon to close the current tab.

You cannot move the tabs to a different position in the workspace as they already cover the entire workspace. You can use the left and right arrows in the top-right corner of the workspace to cycle between all open asset tabs, and the 'X' icon to close the current tab.

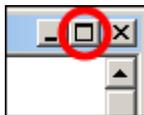
If an asset has any unsaved changes, this is indicated by an asterisk (*) next to the asset name on the tab. When you close a tab that has any unsaved changes, you are first prompted to save or discard these changes.

Switching Between Dialog and Tab Views

You can switch between dialog view and tab view at any time. If you want to see all open assets in tabs rather than dialogs, click the **Maximize** icon of an asset window

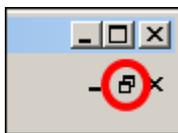
(Figure 5-7). Please note that after you maximize one dialog, *all* assets are displayed in tabs.

Figure 5-7 Maximize Icon



If you want to see all assets in dialogs rather than tabs, click the **Restore Down** icon in the top-right corner of the Designer application window (Figure 5-8), just below the application title bar icons. Please note that after you restore one tab, *all* assets are displayed in dialogs.

Figure 5-8 Restore Down Icon



5.6.2 Views

A site asset may be viewed in various ways, depending on the type of asset:

- **Source view** displays the code that is in the content file associated with the site asset: HTML, XML, JavaScript, Idoc Script, Oracle Site Studio tags, and so on. For more information, see [Section 5.6.3, "Source View."](#)
- **Design view** displays all objects as they are arranged on a site template, both static (for example, a fixed banner graphic) and dynamic (placeholders and fragments). For more information, see [Section 5.6.4, "Design View."](#)
- **Preview** provides an actual view of a site template with all assets in place, as it appears to site consumers in a web browser. For more information, see [Section 5.6.5, "Preview."](#)
- **Form view** provides a dialog in which you can set the properties of a site asset. For more information, see [Section 5.6.6, "Form View."](#)

Figure 5-9 View Tabs in Workspace



Screenshot of three view tabs in the workspace: Source, Design, and Preview. There are left and right arrows in the bottom-left corner to move between the views of the current site asset.

You can use the arrows in the bottom-left corner of the workspace to move between the views of the current site asset. By default, site templates open in design view, while definition assets open in form view. Most other assets (for example, contributor data files and CSS files) open in source view. Native documents open in their associated third-party application.

5.6.3 Source View

Source view displays the code that is in the content file associated with a site asset: HTML, XML, JavaScript, Idoc Script, Oracle Site Studio tags, and so on.

Figure 5–10 Source View (Page Template)

```

<body>
  <p>
    <tbody>
      <tr>
        <td width="20%">
<!--SS_BEGIN_SNIPPET(fragment1,2)--><!--$ssFragmentInsta
</td>

        <td><!--$wcmPlaceholder("Default")--></td>
      </tr>
    </tbody>
  </table><br />
  <br />
</body>
</html>

```

Screenshot of a page template in source view, which shows the actual code (HTML, XML, JavaScript, Idoc Script, Oracle Site Studio tags, and the like).

The information in source view appears as color-coded text:

- Black is used for XML code, HTML tag attributes, and text that appears on the web page.
- Purple is used for HTML tags.
- Blue is used for HTML tag attribute values.
- Green is used for Idoc Script, HTML comments, and code inserted by Oracle Site Studio.

You can change many of these settings, including the typeface, indentation levels, and line wrapping. For more information, see [Section 6.9, "Formatting the Code in Source View."](#)

Source view offers complete control over the site template. If you do not like the way things look in design view, you can always switch to source view to control the exact behavior of the web page. In fact, you may find yourself starting off in design view (see [Section 5.6.4, "Design View"](#)) to create the page and then switching to source view to customize the appearance and behavior of the page.

Most of the sections in this guide assume you are working in design view. Source view instructions are included where appropriate.

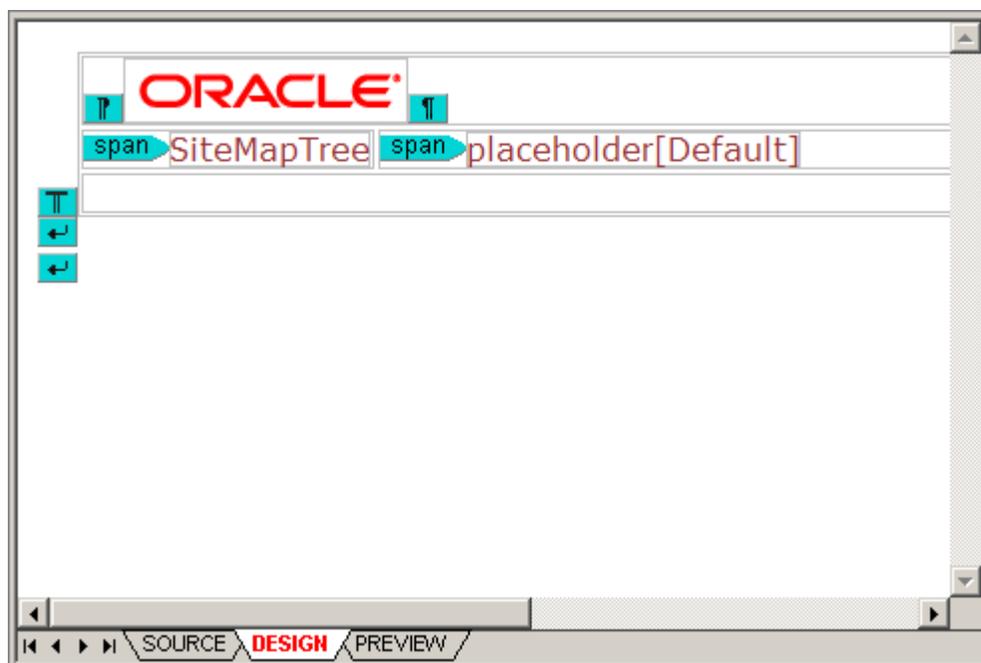
You can edit the text in source view using common editing keystrokes (for example, Ctrl+C and Ctrl+V to copy and paste), the application toolbars (see [Section 5.10,](#)

"Toolbars"), and right-click menus (see [Section 5.6.7, "Right-Click Menus"](#)). You can, for example, add site assets, images, tables, and line breaks as you work in source view.

5.6.4 Design View

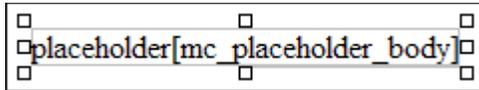
Design view displays all objects as they are arranged on a site template, both static (for example, a fixed banner graphic) and dynamic (for example, placeholders and fragments). Depending on how the template was set up and what positioning methods are used, this view may approximate what the page layout is when viewed in a web browser. You see text, images, colors, hyperlinks, and tables as they appear on the actual web site, but dynamic content (from fragments and scripts) is represented as tags rather than the actual content. This is because most fragments and scripts require server-side scripting and dynamically rendered content. This content can only be seen on the Preview tab (see [Section 5.6.5, "Preview"](#)).

Figure 5–11 Design View (Page Template)



Screenshot of a page template in design view, which displays all objects as they are arranged on a site template.

When you open a site template (page template, region template, or subtemplate) in Designer, it opens in the workspace in design mode, where you can edit the template as needed. The workspace then functions as an editor that offers many common editing features. You can edit the text and objects using common editing keystrokes (for example, Ctrl+C and Ctrl+V to copy and paste), the application toolbars (see [Section 5.10, "Toolbars"](#)), and right-click menus (see [Section 5.6.7, "Right-Click Menus"](#)). You can, for example, add site assets, images, tables, and line breaks as you work in design view. To select an object, simply click it, so it displays selection handles ([Figure 5–12](#)).

Figure 5–12 Selected Object in Design View

For edits that cannot be performed using this interface, you can use the properties pane (see [Section 5.5, "Properties Pane"](#)). The properties pane enables you to fine-tune specific areas of a site template, like background color, page margins, and table width. If you cannot make a specific edit here, you can always edit the actual code in source view (see [Section 5.6.3, "Source View"](#)).

To help you identify where site assets (for example, placeholders and fragments) are located on a site template, Designer inserts asset tags with their names. (You can modify this text for fragments in the Fragment Editor as discussed in [Section 13.12.4, "Adding, Editing, and Deleting Fragment Snippets"](#).)

Designer can also highlight the location of HTML tags in design view. To turn this feature on, click the **Show/Hide HTML Tags** icon ([Figure 5–13](#)) in the formatting toolbar. You can click the down arrow next to the icon to show or hide specific HTML tags.

Figure 5–13 Show/Hide HTML Tags

5.6.5 Preview

Preview provides an actual view of a web page with all assets in place, as it appears in a web browser for consumers. This is useful for previewing site templates and checking how fragments and placeholders are positioned on a template (which may be hard to see in design mode, especially if the page layout is controlled by CSS).

Figure 5–14 Preview (Page Template)

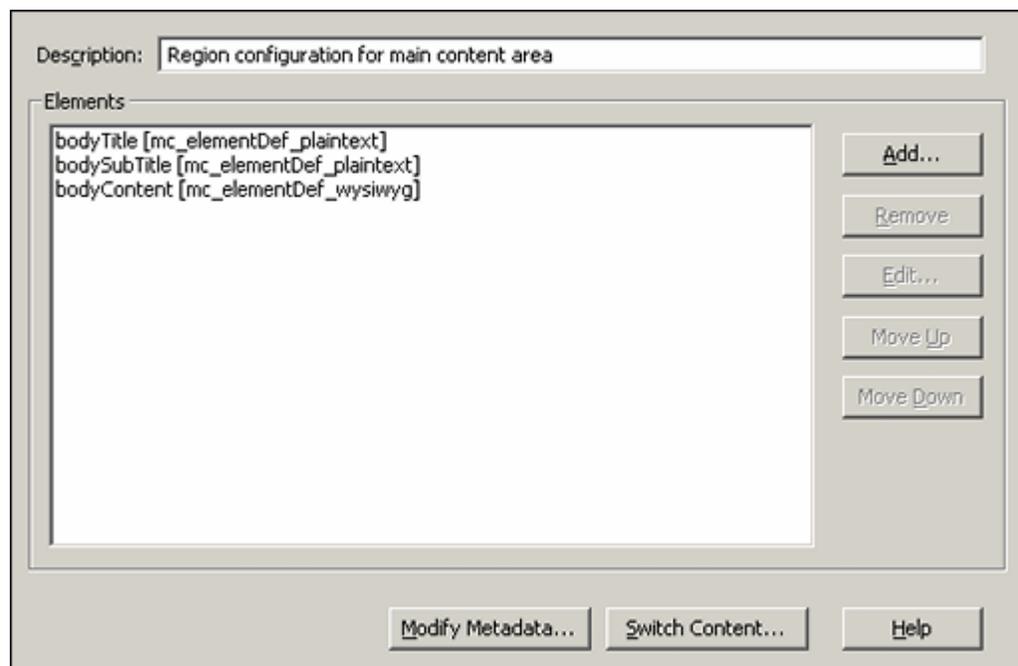
Screenshot of a web page in preview mode, which provides an actual view of the template with all assets in place, as it appears in a web browser for consumers.

Note: Oracle Site Studio requires Internet Explorer 7 or higher to provide previewing capabilities.

5.6.6 Form View

Form view provides a dialog in which you can set the properties of some site assets; for example, element definitions, region definitions, placeholder definitions, and conversion definitions. [Figure 5–15](#) shows an example of a region definition in form view. Most form views have a **Help** button that you can click to see context-sensitive help information for the current form.

Figure 5–15 Form View (Region Definition)

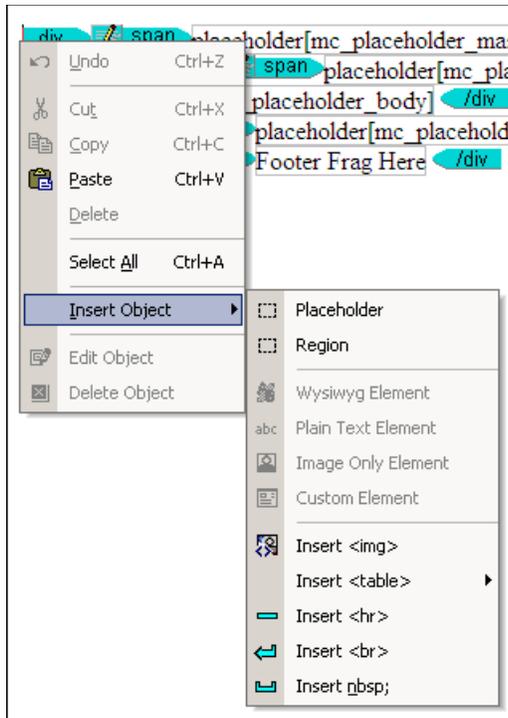


Screenshot of a region definition in form view, which provides an interface where you can manage the region definitions and set their properties.

5.6.7 Right-Click Menus

When you are working in the workspace in source view or design view, you can right-click at any time to see a menu of relevant options ([Figure 5–16](#)). The available options depend on where you are in the workspace, the type of asset you are working on, and what you are doing. In addition to providing several common editing functions (cut, copy, paste, select all, and the like), the right-click menus also typically allow you to insert assets (appropriate to the specific context), or edit or delete a selected asset.

Figure 5–16 Right-Click Menu in Design View

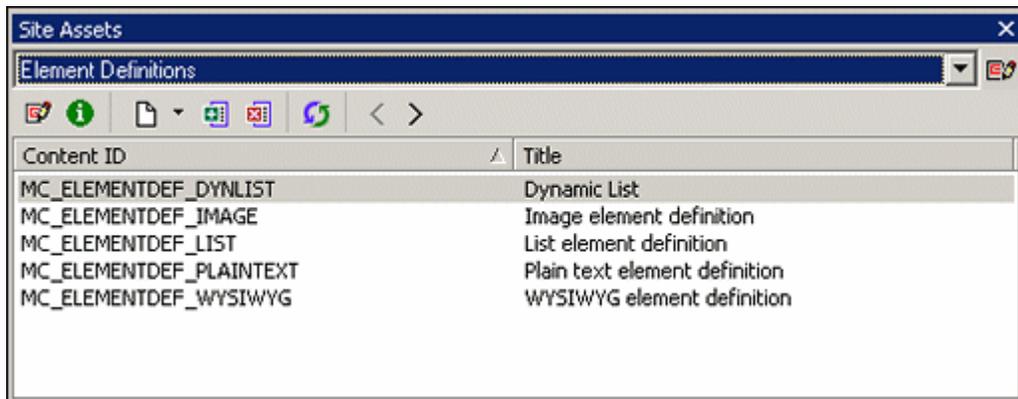


Screenshot of a right-click menu in design view, with common editing options (Copy, Paste, Delete, Select All) and options to insert HTML tags or site assets.

5.7 Site Assets Pane

The site assets pane is used to manage files that are associated with your site. You can use this pane to quickly and easily open, edit, and organize these files. You can show or hide the site assets pane from the **View** menu.

Figure 5–17 Site Assets Pane



Screenshot of the site assets pane, where you can open, edit, and organize all files associated with your site. The various elements of this pane are described in the text that follows this image.

Site Asset List

The site asset list shows all types of assets. This list can be modified as desired; the entries can be arranged and grouped in any way you want by clicking the icon next to the menu. This opens the Site Asset Categories dialog (see [Section A.14, "Site Asset Categories Dialog"](#)), where you can change the way the items in the site assets pane are ordered and grouped.

Toolbar

The site assets pane has a toolbar that provides easy access to several functions:

Icon	Definition
	Opens the Site Asset Categories dialog, where you can customize and even add categories to the site assets pane in Designer. For more information, see Section A.14, "Site Asset Categories Dialog."
	Enables you to edit the selected asset.
	Opens content information for the selected asset.
	Enables you to create assets and copy existing assets.
	Adds the selected asset to the site.
	Removes the selected asset from the site.
	Refreshes the list of items for the current asset category.
	Opens the previous asset category in the list of site assets.
	Opens the next asset category in the list of site assets.

The list of asset categories is stored in the project file of a site, so to maintain the same asset list structure in the menu for each web site, you must edit the list for each site.

Content ID List

After you select a site asset type in the menu, all assets of that type are listed. The content ID is the name of each site asset (as given when the asset was created). When a good naming convention is used, the content ID of the asset can tell you at a glance what the asset is used for. You can sort the list of assets by content ID by clicking the column header.

Title List

Each site asset, when checked into the content server, has its own content ID and a title used for identification. The title is an additional way to help identify the site assets in the site asset pane. You can sort the list of assets by title by clicking the column header.

5.8 Toolbox

The toolbox contains fragments that you can add to page templates, region templates, and subtemplates. Fragments are containers for text, graphics, scripts, and anything else that adds functionality to a web page. You can show or hide the toolbox from the **View** menu.

Figure 5–18 *Toolbox*



Screenshot of the toolbox, which provides access to all available fragments. The fragments are divided into four categories: navigation, dynamic list, static list, and other fragments. The screenshot shows the list of available navigation fragments.

Oracle Site Studio Designer comes with several predefined fragments. There are four categories of fragments: navigation fragments, dynamic list fragments, static list fragments, and other fragments. Each category contains a variety of fragments in multiple scripting languages. You can use any fragment as-is or copy and edit the fragment to suit your needs. You can also create fragments from scratch. For details on the fragments provided with Designer, see [Appendix C, "Sample Fragments."](#)

Adding a fragment to a template is easy. You simply place your cursor in the template where you want the fragment to appear and then click the desired fragment in the

toolbox. Many fragments contain additional options (called parameters) that allow you to customize the appearance of the fragment before adding it to the template.

For more information on fragments, see [Chapter 13, "Working With Fragments."](#)

5.9 Menu Bar

This section contains the following topics:

- [Section 5.9.1, "File Menu"](#)
- [Section 5.9.2, "Edit Menu"](#)
- [Section 5.9.3, "Formatting Menu"](#)
- [Section 5.9.4, "View Menu"](#)
- [Section 5.9.5, "Tools Menu"](#)
- [Section 5.9.6, "Window Menu"](#)
- [Section 5.9.7, "Help Menu"](#)

5.9.1 File Menu

The **File** menu has multiple parts. Many of the items here are ones you would expect to find, such as **Close**, **Save**, **Print**, or **Exit**. However, there are three additional submenus which have many options for working with the structure of the web site, as well as other advanced options.

Many of the items on the File menu are available on the Standard toolbar and Site toolbar. For more information on the Standard toolbar, see [Section 5.10.1, "Standard Toolbar."](#) For more information on the Site toolbar, see [Section 5.10.2, "Site Toolbar."](#)

The File menu contains the following items:

- **Close:** Choose this option to close the active site asset.
- **Save:** Choose this option to save the changes made to the active site asset.
- **Print:** Choose to print the current site asset. This menu item is available only when the active asset is a template. For more information, see [Section 8.2, "Managing Templates as Site Assets."](#)
- **Print Preview:** Choose to view the site asset as it will appear on the printed page. This item is active only when the site asset is in source view. For more information, see [Section 5.6.3, "Source View."](#)
- **Print Setup:** Choose to edit your printer options for this session.
- **Site:** This opens a submenu with administration options that affect the entire web site. For more information, see [Section 5.9.1.1, "Site Submenu."](#)
- **Fragments:** This opens a submenu with options for working with the fragment libraries. For more information, see [Section 5.9.1.2, "Fragments Submenu."](#)
- **Definitions:** This opens a submenu with options for uploading and downloading definition bundles.
- **Exit:** Choose to close Oracle Site Studio Designer.

5.9.1.1 Site Submenu

The **Site** submenu has the following options:

- **Connection Manager:** Opens the Site Connection Manager dialog (see [Section 5.12, "Site Connection Manager"](#)), where you can create, edit, and delete connections to web sites on the content server.
- **Connect:** Establishes a connection with the current web site.
- **Disconnect:** Disconnects from the current web site.
- **Refresh Hierarchy and Fragments:** Refreshes the entire site, which reloads the site hierarchy and fragments for the web site.
- **Recent Sites:** A list of the most recent sites opened in Oracle Site Studio Designer.
- **Advanced:** This opens an additional submenu of advanced items for working with Oracle Site Studio:
 - **Edit Site Addresses:** Opens the Site Addresses dialog. This allows you to specify one or more domain-based or folder-based addresses that point to your web site. For more information, see [Section A.5, "Site Addresses Dialog."](#)
 - **Set Default Link Format:** Opens the Choose Default Link Format dialog. Here you can specify the default link format, starting with choosing between path-based or ID-based, or if changing the format is available in the link wizard. For more information, see [Section A.6, "Choose Default Link Format Dialog,"](#) and [Section 5.13, "Link Wizard."](#)
 - **Define Environment Properties:** Opens the Define Environment Properties dialog. This allows you to determine what properties of the web site are maintained during site replication, and what properties are not replicated. For more information, see [Section A.7, "Define Environment Properties Dialog."](#)
 - **Show Contributor Accessibility Menus:** Toggles the menu bar to a style that works better with screen-reading tools.
 - **Enable Publish Now:** Toggles to include or remove the Publish Now option in the Oracle Site Studio Contributor toolbar. For more information, see [Section 5.20.2, "Publish Now."](#)
 - **Enable Section Level Replication:** Toggles the Ready to Publish field in the Section Properties for all sections of the web site.
 - **Commit Project Changes:** Saves all project changes to the content server.
 - **Update Project Info:** Updates the project information in Designer from the content server.
 - **View Project DOC_INFO:** Opens a browser window displaying the DOC_INFO information of the project file.
 - **View Web Site Objects Report:** Opens a browser window and opens the the Web Site Objects Report. For more information on the Web Site Objects Report, see [Section 14.4, "Viewing a Web Site Objects Report."](#)
- **View:** This opens an additional submenu of items for working with the web site on the Oracle Content Server as well as viewing the web site.
 - **Content Server Portal:** Opens a browser displaying the Oracle Content Server.
 - **Content Server Admin:** Opens a browser window displaying the Oracle Content Administration Server.
 - **Content Server Applets:** Opens a browser window displaying the Administration section on the Oracle Content Server.

- **Content Server Search:** Opens a browser window displaying a search page on the Oracle Content Server.
- **Oracle Site Studio Admin:** Opens a browser window displaying the Oracle Site Studio Administration page on the Oracle Content Server
- **Oracle Site Studio Manage Web Sites:** Opens a browser window displaying the Manage Web Sites section in Oracle Site Studio Administration on the Oracle Content Server.
- **Site Home Page:** Opens a browser window displaying the home page of the current web site
- **Selected Page:** Opens the page highlighted in the hierarchy pane in a browser.
- **Source:** Opens the active asset in the Source tab. For more information, see [Section 8.11.2, "Source View,"](#) and [Section 9.11.2, "Source View."](#)
- **Design:** Opens the active asset in the Design tab. For more information, see [Section 8.11.1, "Design View."](#)
- **Preview:** Opens the active asset in the Preview tab. For more information, see [Section 8.11.3, "Preview."](#)
- **Form:** Opens the active asset in the Form tab. For more information, see [Section 9.11.1, "Form View."](#)

5.9.1.2 Fragments Submenu

The **Fragments** submenu has items to manage fragment libraries. You can also create a new fragment. For more information on fragments, see [Section 13, "Working With Fragments."](#)

- **Select:** Opens the Select Fragment Dialog. Here you can select fragments to use with only the keyboard. The Fragment toolbox requires you use the mouse. For more information, see [Section A.81, "Select Fragment Dialog."](#)
- **New:** Opens the New Fragment dialog. Here you can create a new fragment for the web site, and save it to a library to move to other web sites. For more on creating a fragment, see [Section 13.9, "Creating a New Fragment."](#) For more information on the New Fragment dialog, see [Section A.58, "Properties for Fragment Dialog."](#)
- **Upload Fragment Library:** Opens a dialog to select a stored fragment library on your local computer. For more information, see [Section 13.13.4, "Uploading and Downloading Fragment Libraries."](#)
- **Download Fragment Library:** Opens a browser window listing the fragment libraries on the content server. Select a library to download to the local computer. For more information, see [Section 13.13.4, "Uploading and Downloading Fragment Libraries."](#)

5.9.1.3 Definitions Submenu

The **Definitions** submenu has items to work with uploading and downloading definition bundles. For more information on definition bundles, see [Section 9.12, "Definition Bundles."](#)

- **Upload Definition Bundle:** Opens a dialog to select a stored definition bundle on your local computer. For more information, see [Section 9.12.3, "Uploading Bundles."](#)

- **Download:** Opens another submenu to download a definition bundle based on the type of definition to download. For more on downloading the different definition bundles, see [Section 9.12.2, "Downloading Bundles."](#)
 - **Placeholder Definition Bundle...:** Opens a browser window listing the placeholder definitions on the content server. Select one to continue downloading the bundle.
 - **Region Definition Bundle...:** Opens a browser window listing the region definitions on the content server. Select one to continue downloading the bundle.
 - **Element Definition Bundle...:** Opens a browser window listing the element definitions on the content server. Select one to continue downloading the bundle.

5.9.2 Edit Menu

The **Edit** menu contains items that are used in editing and transforming the text in a document. Many of these functions, such as **Undo**, **Redo**, **Cut**, **Copy**, **Find**, and **Replace** are also used with objects in a document.

Many of the items in this menu are available in the Standard toolbar. For more information, see [Section 5.10.1, "Standard Toolbar."](#)

All items in the Edit menu are unavailable (grayed out) when the active document is in Preview mode. For more information, see [Section 5.6.5, "Preview."](#)

The Edit menu has the following items:

- **Undo:** Reverses the most recent action, undoing it.
- **Redo:** Does the most current action undone by the 'Undo' command.
- **Cut:** Cuts the selected text or object and puts it in the clipboard.
- **Copy:** Copies the selected text or object and puts it on the clipboard.
- **Paste:** Pastes the text or object from the clipboard.
- **Select All:** Selects all text and objects if
- **Find:** Opens a dialog to find an entered text string.
- **Replace:** Opens a dialog to find an entered text string, and replace it with another string.

5.9.3 Formatting Menu

The **Formatting** menu selection of the menu bar contains items that are similar to other software programs. Functions that have to do with formatting text, such as **Bold**, **Italic**, **Numbered List**, **Itemized List**, **Align Center**, and so forth are here.

All items in the Edit menu are unavailable (grayed out) when the active document is in Preview mode. For more information, see [Section 5.6.5, "Preview."](#)

All of the items in this menu are available in the Formatting toolbar. For more information, see [Section 5.10.3, "Formatting Toolbar."](#)

The Formatting menu has the following items:

- **Bold:** Makes the selected text bold.
- **Italic:** Makes the selected text italic.

- **Underline:** Makes the selected text underlined.
- **Remove Formatting:** Removes all bold, italic, and underline formatting in the selected text.
- **Numbered List:** Starts a numbered list of items, starting with 1.
- **Bulleted List:** Starts a bulleted list of items.
- **Increase Indent:** Moves the left-most margin to the right.
- **Decrease Indent:** Moves the left-most margin to the left.
- **Align Left:** Aligns the text to the left margin.
- **Align Center:** Aligns the text to the center of the page.
- **Align Right:** Aligns the text to the right margin.
- **Font Size:** Opens a menu to select a text size from *xx-small* to *xx-large*.
- **Font Face:** Opens a menu to select a font to use on the page.
- **Style:** Opens a menu to select a font style to use.
- **Text Color:** Opens a color picker to select the text color.
- **Background:** Opens a color picker to select the background color (also called "highlighting").

5.9.4 View Menu

The items in the **View** menu are mostly toggles to display windows in the Designer workspace, and toggles to display the workspace toolbars.

The first section of items are used to toggle view (checked) or hide (unchecked) the windows in the workspace. The middle section toggles the workspace toolbars. For more information on the workspace windows, see [Section 5.6, "Workspace."](#) For more information on the different toolbars, see [Section 5.10, "Toolbars."](#)

The final two sections each have a single item:

- **Log File:** Launches the Oracle Site Studio Log File Viewer application. For more information, see [Section 5.11, "Log File."](#)
- **Customize:** Opens the Customize dialog to the Commands tab, where you can determine how the menu bar items are organized. For more information, see [Section A.1.1, "Customize Dialog: Commands Tab."](#)

5.9.5 Tools Menu

The items in the **Tools** menu are used to work with the site currently open in Designer. This section also can contain custom menu items created through the **Customize** item. For more information, see [Section A.1.3, "Customize Dialog: Tools Tab."](#)

The Tools menu has the following items:

- **Update Navigation:** Regenerates the site navigation structures. This will cause menus generated from these structures to also update on the website.
- **Define Custom Section Properties:** Opens the Custom Section Properties dialog, where you can assign custom properties to site sections (in addition to the web site properties such as section ID, label, and default placeholder definition). For more information, see [Section A.9, "Custom Section Properties Dialog."](#)

- **Define Placeholder Definition Mappings:** Opens the Define Placeholder Definition Mappings dialog, where you provide a mapping from a placeholder name to a placeholder definition. For more information on the Define Placeholder Definition Mappings dialog, see [Section A.74, "Define Placeholder Definition Mapping Dialog."](#) For more information on placeholder mappings, see [Section 9.8, "Working With Placeholder Definitions."](#)
- **Compare Changes:** Opens the Compare Changes Feature. For more information, see [Section 12.5.2, "Using the Compare Changes Feature."](#)
- **Create Hyperlink:** Opens the Link wizard. For more information, see [Section 12.4, "Working With Links."](#)
- **Remove Hyperlink:** Removes the selected hyperlink.
- **Show/Hide HTML Tags:** Toggles to show the HTML tags in Design View.
- **Highlight Borderless Elements:** Toggles to show in Design view a thin outline of the elements with content to be shown fully in the final web page, such as placeholders and fragments.
- **HTML Tidy Source:** Applies the HTML Tidy library to the code in Source view. HTML Tidy is a library that is used to clean up HTML source code and have it conform to specified .dtd files.
- **Language:** Selects a language to operate Designer in.
- **Customize:** Opens the Customize dialog to the Tools tab. Here you can add additional custom menu options. For more information, see [Section A.1.3, "Customize Dialog: Tools Tab."](#)

5.9.6 Window Menu

The items in the **Window** menu are similar to those found in other software packages. Here you can select to maximize a window, close it, or select which window among the open workspace windows is the active window.

5.9.7 Help Menu

The items in the **Help** menu are similar to those found in other software packages. Here you can open the help files, or learn more about the version of the Oracle Site Studio Designer program you are currently using.

5.10 Toolbars

Oracle Site Studio Designer has several toolbars that provide easy access to many design functions:

- [Section 5.10.1, "Standard Toolbar"](#)
- [Section 5.10.2, "Site Toolbar"](#)
- [Section 5.10.3, "Formatting Toolbar"](#)
- [Section 5.10.4, "HTML Toolbar"](#)
- [Section 5.10.5, "Table Toolbar"](#)
- [Section 5.10.6, "Contribution Toolbar"](#)
- [Section 5.10.7, "Placeholder Toolbar"](#)

You can show or hide any of these toolbars from the **View** menu. In addition, you can drag and drop the toolbar to any position. If you drag a toolbar to a position outside the toolbar area, it becomes a separate dialog that you can customize by clicking the down arrow in the title bar.

5.10.1 Standard Toolbar

The standard toolbar provides access to several common editing functions such as copying, pasting, and printing.

Figure 5–19 Standard Toolbar



Option	Description
	Saves the open assets.
	Cuts the selected text.
	Copies the selected text.
	Pastes the selected text.
	Cancel the operation that was last performed.
	Does the operation that was last undone by the 'Undo' command.
	Prints the current contents of the active window in the workspace.
	Opens the online help of the Designer application.

5.10.2 Site Toolbar

The site toolbar provides access to several site-related functions such as site connections, content assignment, and navigation.

Figure 5–20 Site Toolbar



Option	Description
	Lists all defined site connections.
	Establishes a connection with the current web site (shown in the list next to this icon).
	Breaks the connection with the current web site (shown in the list next to this icon).

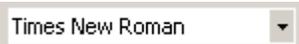
Option	Description
	Opens the Assign Content dialog (see Section A.54, "Assign Content Dialog"), where you can assign content (contributor data file, native document, or subtemplate) to a contribution region on a page.
	Opens the Custom Section Properties dialog (see Section A.9, "Custom Section Properties Dialog"), where you can assign custom properties to site sections (in addition to the web site properties such as section ID, label, and default placeholder definition).
	Regenerates the site navigation structures. This will cause menus generated from these structures to also update on the website.
	Refreshes the entire site, which reloads the site hierarchy and fragments for the web site.
	Opens the Site Connection Manager dialog (see Section 5.12, "Site Connection Manager"), where you can create, edit, and delete connections to web sites on the content server.
	Launches a browser with the currently selected section from the site hierarchy window. Please note that nothing happens if you click this icon with nothing selected in the site hierarchy.
	Compares the current web page with the latest one on the content server, and shows the differences in color-coded text in a new browser window.
	Expands all sections in the site hierarchy to show all assigned primary and secondary pages.
	Collapses all sections in the site hierarchy to hide all assigned primary and secondary pages.

5.10.3 Formatting Toolbar

The formatting toolbar provides access to several text formatting functions such as font size and color, text alignment, lists, and the like.

Figure 5–21 Formatting Toolbar



Option	Description
	Applies the selected font to the text.
	Applies the selected font size to the text.
	Applies the selected color to the text. You can use the down arrow next to the icon to select a text color.
	Applies the selected background (that is, highlight) color to the text. You can use the down arrow next to the icon to select a background color.

Option	Description
	Makes the selected text bold .
	Makes the selected text <i>italic</i> .
	Makes the selected text underlined.
	Removes all formatting from the selected text.
	Inserts a numbered (ordered) list.
	Inserts a bulleted (unordered) list
	Increases the indentation level of the selected text by one level.
	Reduces the indentation level of the selected text by one level.
	Left-aligns the selected text.
	Centers the selected text.
	Right-aligns the selected text.
	Launches the Link wizard (see Section A.18, "Link Wizard"), which enables you to create a hyperlink for the selected text, or edit an existing hyperlink.
	Removes an existing hyperlink from the selected text.
	Shows or hides HTML tags in the workspace. You can use the down arrow next to the icon to show or hide specific HTML tags.
	Shows or hides borders around elements that normally do not have borders.
	Cleans up and reformats the HTML source code, making sure that it complies with the chosen DOCTYPE.

5.10.4 HTML Toolbar

The HTML toolbar provides access to several HTML-related functions such as CSS classes, images, and horizontal lines.

Figure 5–22 HTML Toolbar



Option	Description
	Applies the selected HTML style to the text.
	Inserts an image on the web page at the current cursor position. You are prompted to select an image from the content server.
	Inserts a horizontal line (<hr>).
	Inserts a line break ().
	Inserts a non-breaking space ().

5.10.5 Table Toolbar

The table toolbar provides access to several table-related functions such as adding a table, adding rows and columns, merging and splitting cells, and the like.

Figure 5–23 Table Toolbar



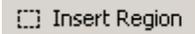
Option	Description
	Inserts a table on the web page at the current cursor position. Click the down arrow next to the icon to choose from a list of predefined tables.
	Deletes the current table.
	Inserts a row to the table above the row the cursor is in.
	Adds a column to the table to the left of the column the cursor is in.
	Splits the table cell where your cursor is located into columns or rows. You are prompted to choose whether the cells should split into columns (that is, vertically) or rows (that is, horizontally).
	Merges the table cell where your cursor is located with the cell to the left.
	Merges the table cell where your cursor is located with the cell above it.
	Merges the table cell where your cursor is located with the cell to the right.
	Merges the table cell where your cursor is located with the cell below it.

5.10.6 Contribution Toolbar

The contribution toolbar provides access to several contribution functions such as adding a region and various types of elements. This toolbar is primary used for legacy sites (that is, sites created in Oracle Site Studio Designer releases before 10gR4).

Figure 5–24 Contribution Toolbar



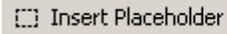
Option	Description
 Insert Region	Inserts a legacy contribution region on the web page at the current cursor position.
	Inserts a legacy WYSIWYG element on the web page at the current cursor position.
 abc	Inserts a legacy plain-text element on the web page at the current cursor position.
	Inserts a legacy image element on the web page at the current cursor position.
	Opens the properties dialog for the selected element, where you can edit its properties.
	Removes the selected item from the web page.

5.10.7 Placeholder Toolbar

The placeholder toolbar provides access to several placeholder-related functions such as adding a placeholder and various types of elements to templates.

Figure 5–25 Placeholder Toolbar



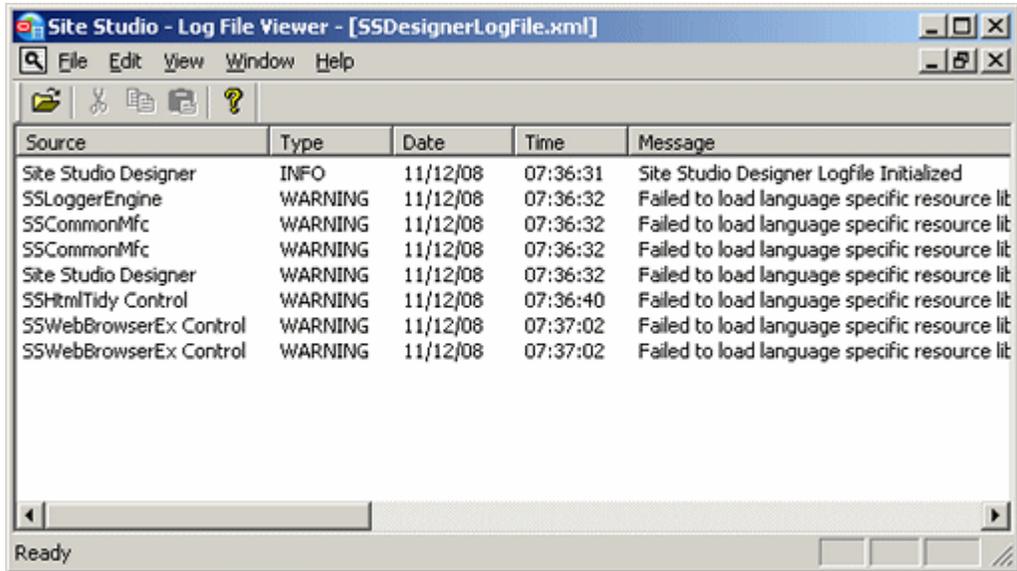
Option	Description
 Insert Placeholder	Inserts a placeholder on the web page at the current cursor position.
	Inserts an element on the web page at the current cursor position.
	Inserts a dynamic conversion on the web page at the current cursor position.
	Opens a dialog where you can assign an element.
	Removes the selected item from the web page.

5.11 Log File

The log file shows status, warning, and error messages that are generated while you are using the Designer application. This may be useful for troubleshooting purposes. You can see the log file at any time by opening the **View** menu and then choosing **Log File**.

The log shows the source of the messages, their type, date, and time, and the actual message texts. You can click any message for a more detailed view. In addition, you can save the current log to a text file by clicking **File**, and then **Save**.

Figure 5–26 Log File



Screenshot of the log file, which shows status and error messages that are generated while you are using the Designer application. The log shows the source of the messages, their type, date, and time, and the actual message texts.

5.12 Site Connection Manager

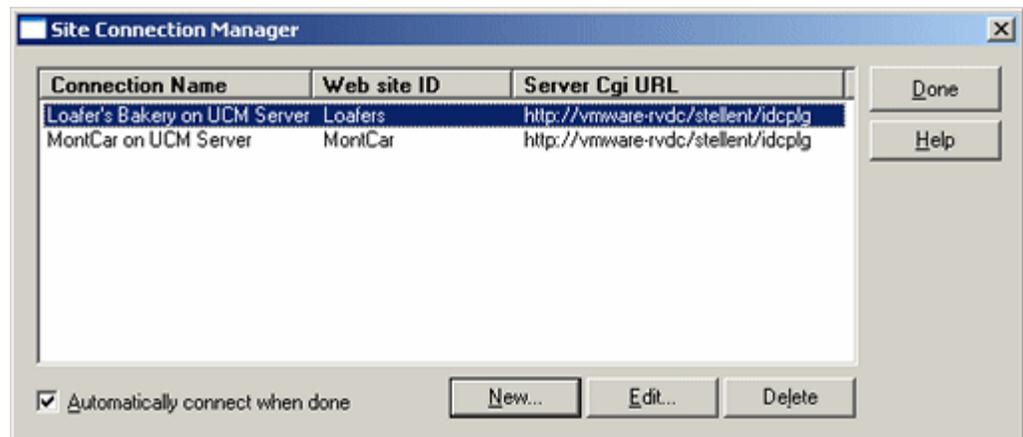
The Site Connection Manager is used to control the connections to Oracle Site Studio web sites on the content server. You can add, edit, and delete site connections. When you connect to a web site, you essentially connect to its project file on the content server. This project file holds all relevant information that enables you to work with the site in Designer. For example, it contains references to all files that comprise the web site and their relationships.

You can open the Site Connection Manager in two ways:

- By opening the **File** menu, selecting **Site**, and then **Connection Manager**.
- By clicking the **Site Connection Manager** icon on the site toolbar (Figure 5–27).

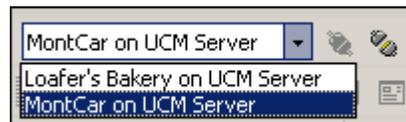
Figure 5–27 Site Connection Manager Icon



Figure 5–28 Site Connection Manager

Screenshot of the Site Connection Manager dialog, where you manage the connections to Oracle Site Studio web sites on content servers. It shows a list of all defined connections, and buttons to add, edit, and delete connections.

When you create a site connection, you specify the Cgi URL address of the content server that stores the web site, the site identifier, and a connection name. Oracle Site Studio Designer then creates a project file for the web site and checks it into the content server with the default metadata for new project files (see [Section 7.3.1, "Specifying the Metadata That is Assigned to a Project File"](#)). When you have created a site connection, you can quickly open your site in Designer using the menu on the site toolbar ([Figure 5–29](#)) or by selecting **File**, then **Site**, and then **Recent Sites**.

Figure 5–29 List of Web Sites in Designer

web sites also appear in the "Web Sites" menu on the content server ([Figure 5–30](#)).

Figure 5–30 List of Web Sites in Oracle Content Server (in Default Tray Layout)

For more information on working with site connections, see [Section 7.1, "Site Connections,"](#) and [Section A.2, "Site Connection Manager Dialog."](#)

5.13 Link Wizard

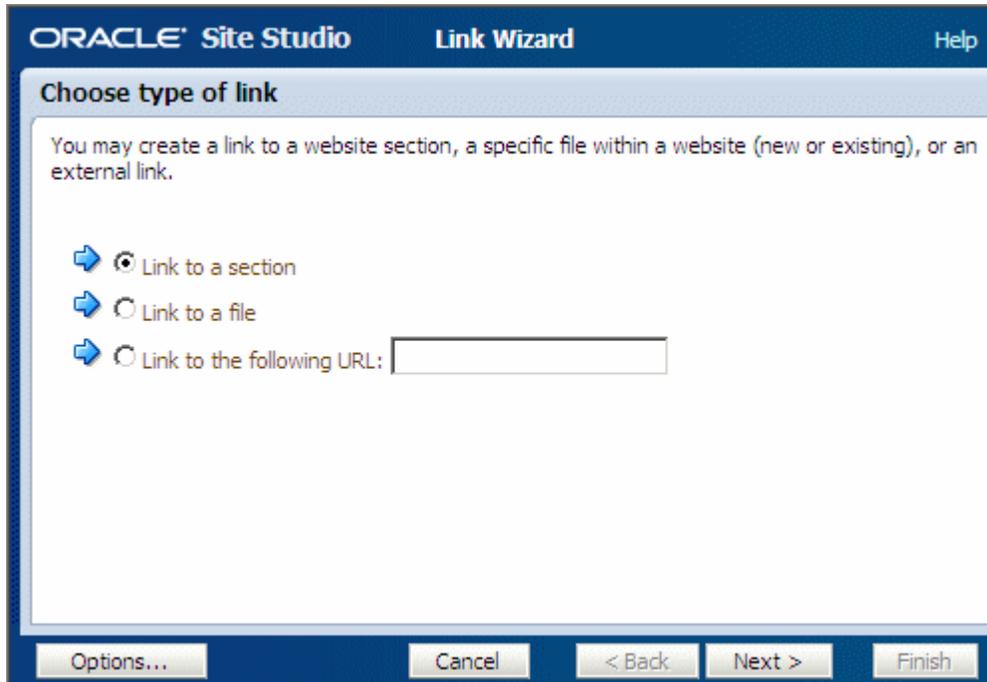
The Link wizard is used to add a hyperlink to text or other page element in page templates, region templates, or subtemplates. You can link to a variety of targets, including web site sections, contributor data files, native documents, and URLs. You can link to items on the current web site, but also to items on other Oracle Site Studio sites on the content server, or to external web sites. Contributors use this same wizard to create hyperlinks in the Contributor application. You, as the site designer, control the options available to them when you set up an element definition (under 'Link Settings'). For example, you decide what kinds of content can be linked to: a new contributor data file or native document, an existing file on the content server, and/or an existing file on your local computer. You may also limit the file formats of native documents that can be linked to (for example, only Microsoft Word documents).

In Designer, with a template open in design view and some text or other page element selected, you open the wizard by clicking the **Create Hyperlink** icon on the formatting toolbar ([Figure 5-31](#)).

Figure 5-31 Create Hyperlink Icon



Figure 5-32 Link Wizard



Screenshot of the Link wizard, which is described in the surrounding text.

The Link wizard consists of multiple steps, depending on the choices you make during the process.

Note: You can step through the wizard faster by clicking the blue arrow to the left of the option you want to use. This automatically selects the option and moves the wizard to the next step.

The first step is to specify the type of hyperlink to create:

- [Link to a Section](#)
- [Link to a File](#)
- [Link to the following URL](#)

You can click **Options** to specify whether the link target should open in a new browser window. After you select the desired link type, the wizard proceeds and presents some additional screens depending on the choices you make.

Link to a Section

If you choose to link to a section, you are asked to specify the site section that the hyperlink points to. You can link to a section on the current web site or a different Oracle Site Studio site on the content server. You select the target section from the site hierarchy.

In addition, you may be asked to select the URL format (path-based or ID-based), depending on whether this was enabled in the wizard (see [Section 12.4.8, "Choosing a Default Link Format"](#)). For more information, see ["URL Formats"](#) on page 5-33.

The final step is to confirm the link before it is actually created (or modified).

Link to a File

If you choose to link to a file, you are first asked to select the type of file that the hyperlink points to. The site designer decides which file type(s) you can link to:

- **New contributor data file:** If you choose to link to a new contributor data file, you are prompted to check the new data file into the content server (using the standard content check-in form). Enter the appropriate content information (metadata) to check the item into the content server. Much of the information is automatically provided. If a field name is marked red, then the field is required (that is, the item cannot be checked in if the field is empty). Please note that the new contributor data file is empty until a site contributor edits it in the Contributor editor.

You are also prompted to specify the web site section in which you want the hyperlinked data file to appear. For more information, see ["Target Sections"](#) on page 5-33.

In addition, you may be asked to select the URL format (path-based or ID-based), depending on whether this was enabled in the wizard (see [Section 12.4.8, "Choosing a Default Link Format"](#)). For more information, see ["URL Formats"](#) on page 5-33.

The final step is to confirm the link before it is actually created (or modified).

- **New native file:** If you choose to link to a new native document, you first must select the file format (for example, Microsoft Word). You, as the site designer, decide what file formats are allowed for new native documents.

Next, you are prompted to check the new native document into the content server (using the standard content check-in form). Enter the appropriate content information (metadata) to check the item into the content server. Much of the information is automatically provided. If a field name is marked red, then the field

is required (that is, the item cannot be checked in if the field is empty). Please note that the new native document is empty until a site contributor edits it in its associated third-party application.

You are also prompted to specify the web site section in which you want the hyperlinked native document to appear. For more information, see "[Target Sections](#)" on page 5-33.

In addition, you may be asked to select the URL format (path-based or ID-based), depending on whether this was enabled in the wizard (see [Section 12.4.8, "Choosing a Default Link Format"](#)). For more information, see "[URL Formats](#)" on page 5-33.

The final step is to confirm the link before it is actually created (or modified).

- **Existing file from server:** If you choose to link to an existing file on the content server, you are taken to the standard search results page on the content server, showing all content files that are available for you to choose. You, as the site designer, decide what criteria are used for the search query. Click the **Select** button next to the content item you want to link to.

You are also prompted to specify the web site section in which you want the hyperlinked file to appear. For more information, see "[Target Sections](#)" on page 5-33.

In addition, you may be asked to select the URL format (path-based or ID-based), depending on whether this was enabled in the wizard (see [Section 12.4.8, "Choosing a Default Link Format"](#)). For more information, see "[URL Formats](#)" on page 5-33.

The final step is to confirm the link before it is actually created (or modified).

- **Existing local file:** If you choose to link to an existing local file (that is, on your own computer), you are prompted to check the file into the content server (using the standard content check-in form). Much of the information is automatically provided. If a field name is marked red, then the field is required (that is, the item cannot be checked in if the field is empty). Click the **Browse** button next to the Primary File field to navigate to the file on your computer and select it. Also, make sure to specify a title.

You are also prompted to specify the web site section in which you want the hyperlinked file to appear. For more information, see "[Target Sections](#)" on page 5-33.

In addition, you may be asked to select the URL format (path-based or ID-based), depending on whether this was enabled in the wizard (see [Section 12.4.8, "Choosing a Default Link Format"](#)). For more information, see "[URL Formats](#)" on page 5-33.

The final step is to confirm the link before it is actually created (or modified).

- **Current item:** Choose this option if you want to continue to link to the selected content file, but you want to change the hyperlink properties (for example, its target section on the site).

Link to the following URL

If you choose to link to a URL (web address), you can specify that URL in the field next to this option. You cannot proceed in the wizard until you provide a URL. This is often an external URL, such as a web site on the Internet, but may also be useful to link to a file on the content server that is not used for contribution, such as a PDF file, media file, or zip file.

The final step is to confirm the link before it is actually created (or modified).

Target Sections

One of the wizard steps enables you to specify the web site section in which you want the selected data file to appear (the "target section"). This enables you to control where a particular file appears on the web site regardless of where it is actually stored on the site. For example, if you have a product description stored in the "Products" section of the site, you can create a link to it from the "Support" section and then specify that the document actually appears in the "Support" section when the hyperlink is clicked. You can choose from the following options:

- **Use default Web site section metadata:** If you choose this option, the hyperlinked content file opens in the web site section as currently specified in the Web Site Section metadata field for the content file. If contributors are allowed to select the URL format of links, they must use an ID-based URL format.
- **Choose a Web site section:** If you choose this option, the hyperlinked content file opens in the web site section of your choice (in effect overriding the default target section of the content file). The section may be in the current web site or another Oracle Site Studio site on the content server.
- **Link to the Content Item's URL:** If you choose this option, the file displays exactly as it is stored on the content server rather than within the framework of the web site. This is useful for linking to native documents such as PDF files.

URL Formats

You may be prompted to choose the URL format of the link, depending on whether contributors are allowed to make this decision. (You can set this option in Designer by selecting **File**, then **Site**, then **Advanced**, and then **Set Default Link Format**.) The following URL formats are available:

- **Path Based URL:** The link contains a path to the target location. You have two choices for this type of link:
 - **Absolute Path:** Generates a full path; for example
`<!--$ssServerRelativeSiteRoot-->products/index.htm` (where
`"<!--$ssServerRelativeSiteRoot-->"` gets replaced with the path to the root of the web site).
 - **Relative Path:** Generates a relative path instead of a full path; for example
`../products/index.htm`.

Please note that this option is not available if you chose to use the default web site section for the content file as the target section.

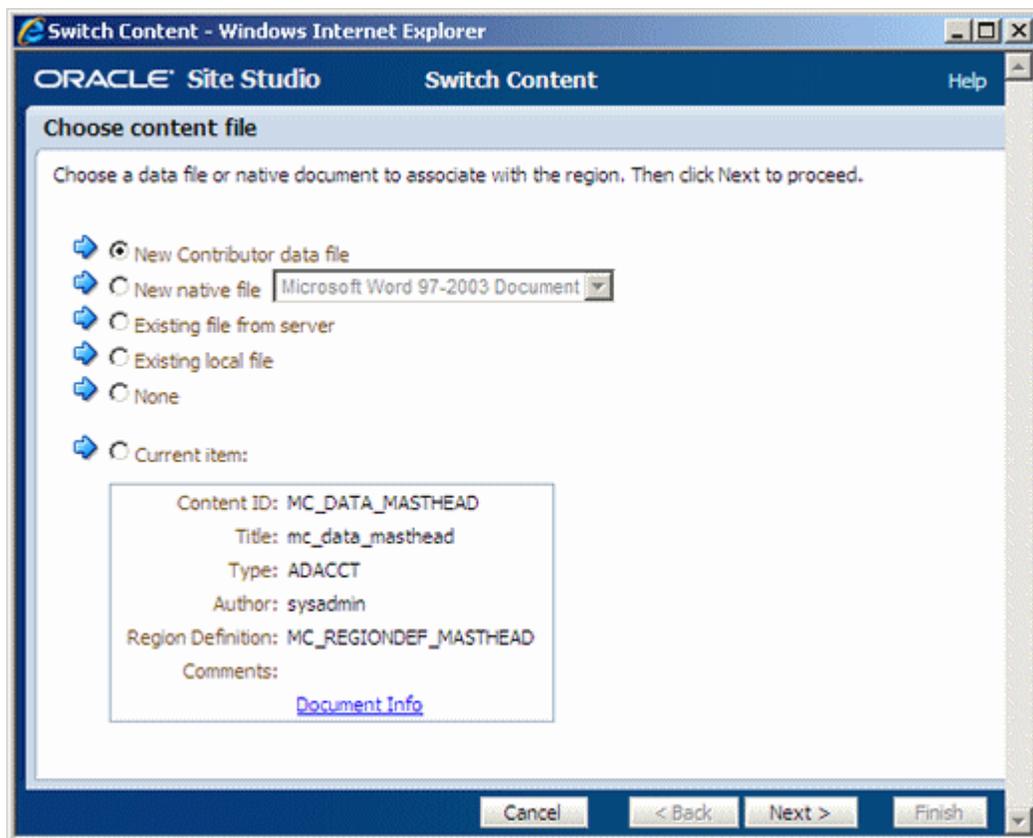
- **ID Based URL:** The link contains the coded identity of the target location rather than the path-based name. You have three choices for this type of link:
 - **Client Side Script Format:** Uses client-side JavaScript to construct a link to the target location; for example, `javascript:nodelink('10024');`.
 - **Server Side Script Format:** Uses server-side Idoc Script to construct a link to the target location; for example, `<!--$wcmUrl('nodelink','10024')-->`.
 - **URL Token Format:** Uses a redirect on the server to construct a link to the target location; for example, `ssNODELINK/10024`.

5.14 Switch Content Wizard

The Switch Content wizard is used to change the content assigned to a contribution region on a web page (or assign content if none was assigned yet). Contributors use this same wizard to assign or switch content in the Contributor application. You, as the site designer, control the options available to them when you set up a region definition. For example, you decide what kinds of content can be assigned: a new contributor data file or native document, an existing file on the content server, and/or an existing file on your local computer. You may also limit the file formats of native documents that can be assigned (for example, only Microsoft Word documents). In addition, you may allow the association of content with a contribution region to be removed altogether.

In Designer, you open the Switch Content wizard from a web page that opens in the Assign Content dialog (see [Section A.54, "Assign Content Dialog"](#)), which, in turn, is launched by right-clicking an assigned primary or secondary page of a section in the site hierarchy and choosing **Assign Content**. The page is essentially displayed in contribution mode, and you can right-click the menu in a contribution graphic to switch the content of the associated placeholder (or assign it if no content has yet been assigned).

Figure 5–33 Switch Content Wizard



Screenshot of the Switch Content wizard, which is described in the surrounding text.

The Switch Content wizard consists of multiple steps, depending on the properties of the current contribution region and the choices you make during the process.

Note: You can step through the wizard faster by clicking the blue arrow to the left of the option you want to use. This automatically selects the option and moves the wizard to the next step.

Selecting a Region Definition or Subtemplate

Depending on how contribution region was set up, you may first be prompted to select the region definition or subtemplate that the region content should be based on.

If you select a **region definition**, you associate the contribution region directly with a contributor data file or native document (to be selected later in the wizard). Its contents open in the contribution region in accordance with the selected region definition and its associated region template. You, as the site designer, control what region definitions are available for the contribution region (in the placeholder definition).

If you select a **subtemplate**, you associate the contribution region with a subtemplate that defines what the region looks like and typically divides it into smaller contribution regions (each of which needs separate content assigned to it). You, as the site designer, control what subtemplates are available for the contribution region (in the placeholder definition).

Selecting the Content

Depending on your wizard choices, you may be prompted to select the type of content file that is associated with the contribution region. The available content file types depend on what was made available in the region definition. Select an option:

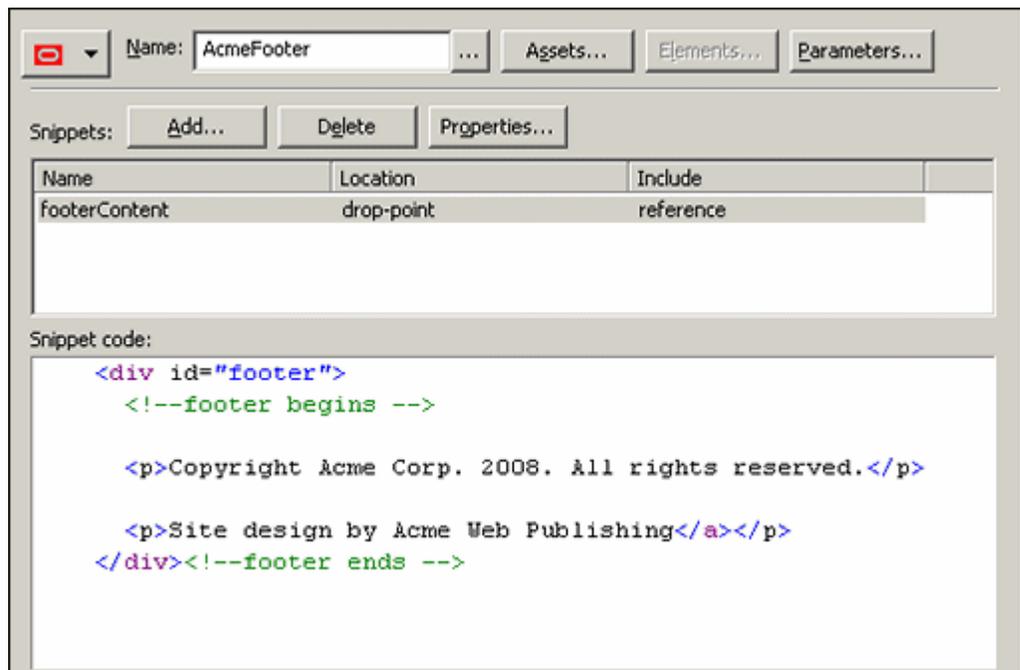
- To use a new contributor data file, select **New Contributor data file**. You are taken to a content check-in form, where you can provide the metadata for the contributor data file and check the file into the content server. The file is empty until you (or another contributor) edits it in the Contributor editor.
- To use a new native document, select **New native file** and choose the file format. (The available file formats are set in the region definition.) You are taken to a content check-in form, where you can provide the metadata for the native document and check the file into the content server. All required metadata fields are marked red. The file is empty until you (or another contributor) edits it in its native application.
- To use an existing file on the content server, select **Existing file from server**. You are taken to a search results page, which lists all items on the content server that you can choose from. (The criteria for the search query are set in the region definition.) Select the file you want to use by clicking its associated **Select** button.
- To use an existing file on your computer, select **Existing local file**. You are taken to a content check-in form, where you can provide the metadata for the file and check the file into the content server. All required metadata fields are marked red. Use the **Browse** button next to the **Primary File** field to navigate to the file on your computer to use and select it.
- To remove the existing association of a content file with the contribution region, select **None**. This means that the contribution region no longer has a content file assigned to it.

5.15 Fragment Editor

The fragment editor is used to create and edit fragments. To edit a fragment, simply right-click the fragment in the toolbox (see [Section 5.8, "Toolbox"](#)) and choose **Edit** or **Copy and Edit** (for read-only fragments). This opens the fragment in the fragment editor. (You may first see a fragment properties dialog, where you must provide some information before you can edit the fragment.) To create a fragment, select **File**, then select **Fragments**, and then **New**.

The fragment editor may display as a dialog or a tab. For more information, see [Section 5.6.1, "Dialogs vs. Tabs."](#)

Figure 5–34 *Fragment Editor*



Screenshot of the fragment editor, with the code of a simple copyright fragment, buttons to add and delete fragments, and set the fragment parameters.

Fragments can exist in many forms, from a simple copyright statement to a complex series of scripting functions. You can turn just about any piece of code into a fragment. The Fragment Editor helps you manage this code and the individual parts that comprise a fragment, including the following:

- **Properties:** The properties of a fragment are its identifying characteristics such as its name, type, and scripting language.
- **Snippets:** Snippets are the individual pieces of code (which could simply be text) that comprise a fragment. This code can be created and edited in source view or design view in the Fragment Editor (similar to source and design view of a site template). You must know the correct scripting syntax if your fragment contains scripting code.
- **Assets:** Assets are files that are referenced from within a fragment. A graphic, cascading style sheet, or JavaScript file are all commonly used assets. After you add an asset to a fragment, it becomes part of that fragment.

- **Elements:** Elements can be added to a fragment that is used to generate a static list. When elements are used this way, contributors can edit the list, add rows, delete rows, and more.
- **Parameters:** Parameters are the attributes that govern the appearance and behavior of a fragment. You can specify your own set of parameters when you create a fragment and then choose from those parameters when you add the fragment to a page template.

For more information on the fragment editor, see [Section 13.12, "Using the Fragment Editor."](#)

5.16 Contributor Data Files and Native Documents

Contributor data files and native documents are used to store content submitted to the site by a contributor. Both files can be assigned directly to a contribution region on a primary page or added as the target of a hyperlink or dynamic list, where they appear on a secondary page.

The two files are different, however, in how you open and edit them:

- **Contributor data file:** A contributor data file is an XML file generated by Oracle Site Studio and intended to be opened and edited using the Contributor application. The file is immediately updated and added to the web page.
- **Native document:** A native document is a third-party file that is viewed and edited using the application originally used to create it (for example, Microsoft Word is used for Word documents). When the file is closed, it is converted into a web page (using Dynamic Converter) and then made available to the web page.

To use these files on your site, you must create an editable area (a 'contribution region') on a page template and then assign the data file or native document to the placeholder. If you are using data files, you also must add one or more elements to each region definition. Each element appears as a field in Contributor where a user can add and edit content. You, as the site designer, control the editing options available to the contributor within each element (in its element definition).

After you have set up the contribution region, you assign one of these files to the region using the Assign Content dialog (see [Section A.54, "Assign Content Dialog"](#)). This is available at any time by clicking the **Assign Content** icon ([Figure 5–35](#)).

Figure 5–35 Assign Content icon



When you first open the Assign Content dialog, you see the site hierarchy on the left and the fully rendered page template with one or more contribution regions on the right. Each contribution region has its own contribution graphic, which shows the name of the contribution region, a **Status/Edit** icon, and a **Menu** icon ([Figure 5–36](#)).

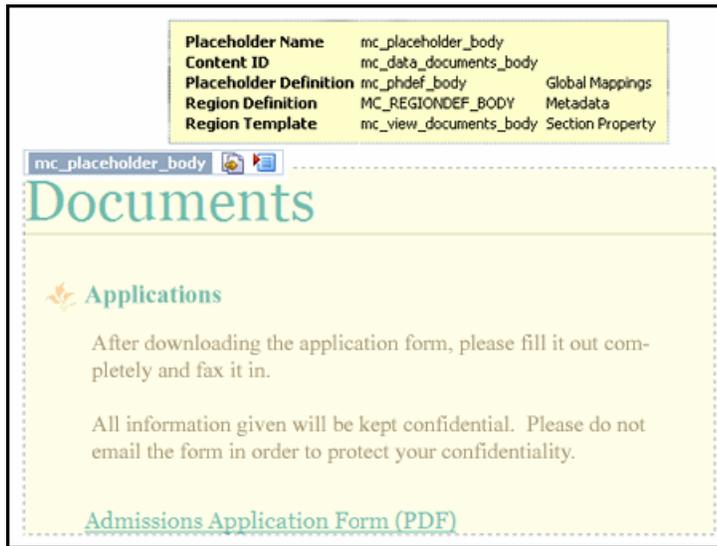
Figure 5–36 Contribution Graphic



Contribution graphic, with region name, edit icon, and 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. In addition, a tooltip appears that provides useful site design information for the contribution region (such as the associated placeholder name and definition, the content ID of the associated content file, and the associated region definition and template) (Figure 5–37).

Figure 5–37 Contribution Region Marked on Web Page With Tooltip



Screenshot of a contribution region on a web page that is marked by a yellow box, with a tooltip that provides useful design information about the contribution region.

If not all contribution regions on a page in a site section have content assigned to them, the folder icon of that section in the site hierarchy shows a red 'x' (Figure 5–38).

Figure 5–38 Red X Icon



You can specify a new data file or native document or reuse an existing one for each region. You should specify a new file if you want new and unique content to display on that page on the web site and reuse an existing file if you want the same content to display on multiple pages on the web site (similar to the decision that goes into reusing other site assets, such as page templates).

Note: Contributor data files and native documents are stored in the content server like any other content item. As such, you can choose different metadata or security for each file and consequently, limit what a contributor sees based on his or her login credentials.

Allowing Contributors to Create Files

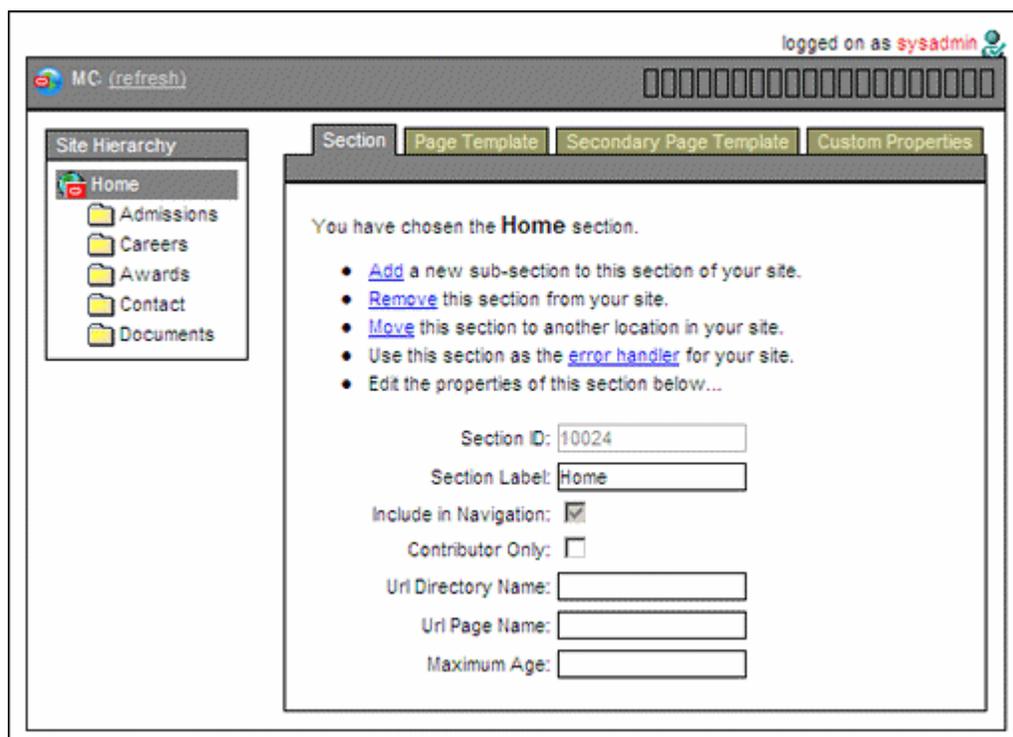
In addition to assigning contributor data files or native documents to a region, you can allow contributors to create their own data files and native documents on the web site.

This adds a whole new dimension to your site, allowing the site to grow without your having to create sections and page templates every time a piece of content is created. To do this, you must allow contributors to add files to the site (an allowed action in an element definition) and then set up areas on the site where those files can appear (a replaceable region on a secondary page).

5.17 Oracle Site Studio Manager

When a web site has been created and deployed using Oracle Site Studio Designer, the site management responsibilities can be handed over to users who manage the site. They do this using Oracle Site Studio Manager, which is a web-based application.

Figure 5–39 Oracle Site Studio Manager Application



To use Manager, you browse to a web page that has the manager feature enabled (see [Chapter 16, "Setting Up Manager"](#)). Depending on how it is implemented, Manager may display at all times or it may only display in contribution mode. You can open Manager from each page or from a designated section of the site.

In Manager, you can modify the site hierarchy, change the page template associated with a section, modify the properties of a section, and so on. The manager of the site may choose to work exclusively in Manager or both Manager and Contributor to effectively manage the site.

When you add Manager and Contributor to your site, you must consider how the two applications are used and how they might be used together by managers and contributors. You should educate them regarding their roles and responsibilities.

For more information on using Manager, see *Managing Oracle Site Studio*.

5.18 Oracle Site Studio Contributor

After a web site has been created and deployed in Oracle Site Studio Designer, it can be handed over to the users who add and update content on the site. These users are called contributors, and they update the site using the Contributor application.

The designer of the site can also use Contributor, and may frequently do so to test the site contribution, to populate the site with sample pages, or to fix pages that have become broken.

To start Contributor, you visit a web page that contains an editable contribution region (which was set up in Designer) and then change the page view to contribution mode (you may be prompted to log onto the web site). You can then click any contribution graphic on the page and start editing that page in Contributor.

The following topics provide a brief introduction to Contributor:

- [Section 5.18.1, "Launching Contributor"](#)
- [Section 5.18.2, "Contributor Editor"](#)
- [Section 5.18.3, "Contributor and Workflows"](#)

5.18.1 Launching Contributor

To open the Contributor application, browse to the web page containing a contribution region and enable contribution mode using a combination of keystrokes on the keyboard (the default is Ctrl+Shift+F5). You are then prompted to enter your login credentials.

You can tell that you are in contribution mode by a thin Contribution Mode banner along the top of the screen. You also see one or more contribution graphics (Figure 5-40) on the page, one for each placeholder.

Figure 5-40 Contribution Graphic



Each contribution graphic marks an editable area on the web page that can be opened and edited with Contributor or a third-party application (if a native document is used). The name of the placeholder is listed first, followed by the **Edit** icon then the menu icon. To edit, click the **Edit** icon or click the **Menu** icon and choose **Edit**.

Please note the following:

- The number of contribution graphics may vary, depending on the number of editable areas on a page and the contributor's login credentials (if the designer set up different security for each file on the site).
- Different icons may appear within the contribution graphic, meaning different things. If content has not been assigned to the placeholder, then the graphic has a small green '+' symbol (Figure 5-41).

Figure 5-41 Contribution Graphic Showing That Content Has not Been Assigned



If the contributor has access to switch the region content, then the graphic displays a small yellow arrow (Figure 5-42).

Figure 5–42 Contribution Graphic Showing That Content Can Be Switched

If the content is in workflow, this is represented by a green gear icon on the left side of the contribution graphic (Figure 5–43).

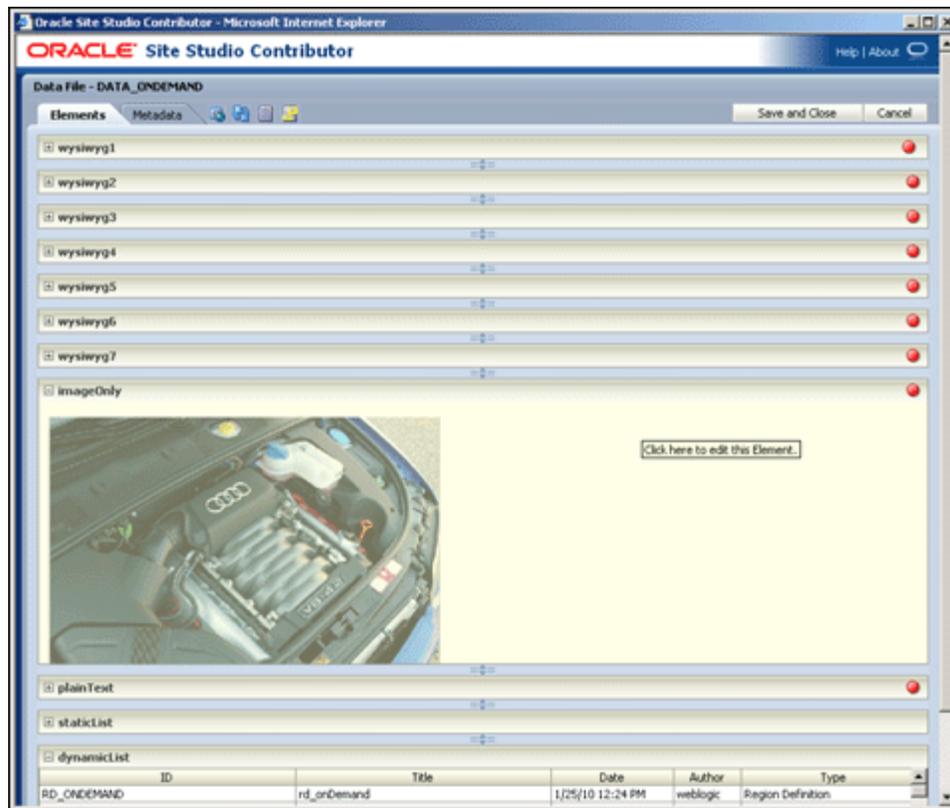
Figure 5–43 Contribution Graphic Showing Workflow

- For information on how to change the default keystroke combination (Ctrl+Shift+F5), see *Managing Oracle Site Studio*.
- To know which web pages contain contribution regions and of those, which ones should be edited, the designer must notify the contributor, perhaps by forwarding the URL of the web page or using Workflow.

5.18.2 Contributor Editor

After you choose to edit a contribution region on a web page, the Contributor editor launches in a browser popup window (Figure 5–44), and you can start editing the content in the current contribution region. Please note that this browser popup window may be suppressed by popup-blocking software, so you must configure that software to allow pop-ups on your site. Also, 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.

Figure 5–44 Contributor Editor



Screenshot of the main Contributor editor window, which is explained in the surrounding text.

The Contributor interface is very similar to the interface of a word processing program. You can add, remove, and edit text, images, hyperlinks, and more. There is even a list option that enables you to manage simple or complex lists. The appearance and behavior of Contributor varies depending on how each contribution region and element is set up in Designer. As the designer of the site, you can set up and modify contribution regions so that the Contributor interface is plain and simple or advanced and powerful.

The main editor window contains various elements:

- **Page header:** includes the name of the contribution region being edited.
- **Data file:** shows the content ID of the data file associated with the editable contribution region on the content server.
- **Elements tab:** provides the editing environment for all elements in the contribution region. Please note that *all* elements in the associated data file are shown, even if only some of them are actually displayed in the contribution region being edited. (The other information may be used elsewhere on the web site, so editing that information may affect other pages on the site.)
- **Metadata tab:** shows the content information (metadata) for the data file associated with the editable contribution region (only if the site designer opted to make the metadata available for the contribution region).

- **Contribution toolbar:** provides a set of functions that apply to the contribution region as a whole. You can save or preview the data changes, refresh the current view, open a usage report for the data file, and view the changes made to the web page.
- **Editing areas:** one for each element in the contributor data file associated with the contribution region on the web page. These editing areas are where you type in your text and format your document.
- **Element toolbars:** provide editing functionality relevant to the type of content you are editing. The site designer controls what editing options are available for each element type.

5.18.3 Contributor and Workflows

In addition to contributing content to the site, a contributor may also be responsible for reviewing and approving content. This could also be the sole responsibility of a reviewer in your organization. The process of reviewing and approving content is called workflow.

You create a workflow in Oracle Site Studio using the existing workflow functionality in the content server. If you are familiar with this, you should be able to quickly adopt the workflow process on your site. Workflow can be enabled for the content assigned to a contribution region on a web page. When a contributor adds a new data file or native document to that web page, the content automatically enters a workflow (preventing the page from going "live" without going through an approval process).

To participate in a workflow, reviewers click a link in a workflow e-mail message that takes them to the web page that must be reviewed (the page is in contribution mode). They can open the menu in the contribution graphic that has the workflow icon (Figure 5-45) and choose **Approve Document**, **Reject Document**, or **Edit**.

Figure 5-45 *Contribution in Workflow Icon*



If the reviewer chooses **Approve Document**, the next reviewer in the workflow is notified with a similar e-mail message. If the reviewer is the last reviewer in the workflow, clicking **Approve Document** makes the content appear on the actual, "live," web site.

If the reviewer chooses **Reject Document**, the Oracle Content Server rejection page opens, where the review must supply details on why it was rejected. Additionally, an e-mail is sent to the previous reviewer in the workflow. Clicking **Edit** opens Contributor so that the content can be edited. For more information, see [Section 15.3, "Workflow Experience for Contributors,"](#) and *Using Oracle Site Studio Contributor*.

The complete workflow process depends on how you set up workflow on the content server. For more information, see [Chapter 15, "Using Workflows,"](#) and the Oracle Content Server documentation.

5.19 Oracle Site Studio Administration Page in Oracle Content Server

The Oracle Site Studio Administration page is created when you install the Oracle Site Studio component on the content server, and it is accessible from the Oracle Content Server Administration page.

You use this page to perform several administrative tasks for all of your web sites (some of which you can also do in Designer). Using the Oracle Site Studio Administration page, you can view information about your web sites, start and stop web sites, replicate web sites, upgrade web sites, and much more.

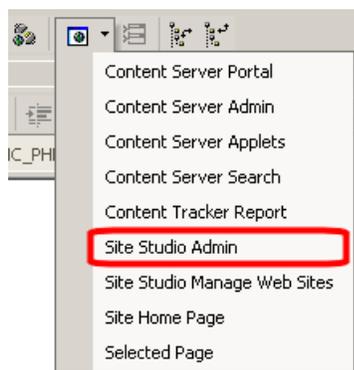
This page has the following options:

- **Manage Web Site:** Opens the Manage Web Sites page, where you can view each site, update navigation, view a site report, stop and start the site, and reload a site.
- **Manage Web Site Addresses:** Opens the Manage Web Site Addresses page, where you can add domain addresses that point to your web site.
- **Manage Fragment Libraries:** Opens the Manage Fragment Libraries page, where you can deploy fragment libraries, configure JSP support, and compress the Contributor code.
- **Set Default Project Document Information:** Opens the Set Project Default Document Information page, where you can specify the default metadata that is assigned to project files (created in Designer).
- **Set Default Web Asset Document Information:** Opens the Set
- **Manage Site Replication:** Opens the Manage Site Replication page, where you can replicate your site from one content server to another.
- **Backup and Restore:** Opens the Backup and Restore page, where you can create a backup of your web site and restore from a backup of your web site.
- **Oracle Site Studio Publisher:** Opens the Oracle Site Studio Publisher page, where you can manage the scheduled publication of an Oracle Site Studio web site to a static web site. For more information, see *Using Oracle Site Studio Publisher*.
- **General Component Information:** Opens the General Component Information page, where you can view useful information about your site, including the version, Oracle Site Studio-specific metadata, directory paths, and so on. Additionally, you can enable or disable the Contributor console from this page.

For more information on each of these options, see *Using Oracle Site Studio Publisher*.

In addition to going through the Oracle Content Server user interface, you can also conveniently open the Oracle Site Studio Administration page by clicking the down arrow next to the **View Pages** icon on the site toolbar in Designer and choosing **Oracle Site Studio Admin** (Figure 5-46). This opens the Oracle Site Studio Administration page on the content server hosting the site that you are currently connected to in Designer.

Figure 5-46 Oracle Site Studio Admin Option in View Menu



Screenshot of the view menu in the site toolbar, with the Oracle Site Studio Admin option highlighted.

5.20 Oracle Site Studio Publisher

Oracle Site Studio Publisher is a component installed on the Oracle Content Server (just as the Oracle Site Studio component is installed) that is used to crawl an Oracle Site Studio web site and transfer it from an Oracle Content Server environment to a pure web server environment that is not running an Oracle Content Server instance. This process is referred to as publishing.

Oracle Site Studio Publisher creates a static snapshot of a dynamic site by traversing all the links in a web site (visiting all of the linked pages) and downloading a copy of each page and all of the resources (images, flash movies, and so on) on each page. Your entire web site, including the content of queries, layout pages, fragments, contributor data files, and native documents, are then copied and published to the new server.

This section contains the following topics:

- [Section 5.20.1, "Oracle Site Studio Publisher in Oracle Content Server"](#)
- [Section 5.20.2, "Publish Now"](#)

5.20.1 Oracle Site Studio Publisher in Oracle Content Server

Oracle Site Studio Publisher is run from the Oracle Content Server. Once installed, you can create any tasks you need to crawl any sites created with Oracle Site Studio. It is not necessary that Publisher is located on the content server storing the web site that you want to publish.

Figure 5–47 Main Oracle Site Studio Publisher Screen

Running						
<div style="display: flex; justify-content: space-around;"> Pause Publishing Add New Task Refresh </div>						
Description	Priority	Status	Next Run	Last Run	Progress	Actions
Weekly Tuesday Automated Publish	normal	waiting	Nov 17, 2009 7:00:00 AM			
runondemand	high	waiting		Nov 16, 2009 2:43:19 PM	Completed	

The main Publisher interface is a type of dashboard, listing all of the publishing tasks, their status including when they are scheduled to run and most recently ran, and an actions menu to modify individual tasks, or even run them immediately ("on demand.")

This window contains the information on available publishing tasks:

- The top of the screen lists the **status** of Publisher, if it is running or paused.
- **Pause Publishing** suspends all tasks from running according to schedule. If Publisher is paused, then the button becomes **Resume Publishing**.
- **Add New Task** is used to add a new publishing task.
- **Refresh** redisplay all task statuses.
- **Description** is the name you have given the task.

- **Priority** shows if the task runs at a high or a normal priority.
- **Status** displays if the task is Waiting, Pending, Running, Finished, or Expired.
- **Next Run** is the scheduled date and time the task will run.
- **Last Run** is the date and time of the completion or failure of the most recent task.
- **Progress** displays the current progress of the task.
- Click **Actions** to open a menu and select one of the following actions for the task:
 - **Edit** opens a screen to edit the task parameters.
 - **Run** puts the task in the queue to run according to the schedule set for the task.
 - **Stop** appears when a task is running. Select to stop a task.
 - **Delete** deletes the task.
 - **View Info** opens a screen to view the information about previous times the task has run. This will not appear on tasks that have never run.
 - **View Logs** opens the screen to view the log file for the task.

See *Using Oracle Site Studio Publisher* for more information on using Publisher.

5.20.2 Publish Now

Publishing is the term used to describe Oracle Site Studio Publisher crawling a web site and creating a static web site to a pure web server environment. Depending on the size of the web site, publishing may take quite a while, and this is why the full site is often published only periodically. In these cases, some time may pass before a changed web page is reflected on the "live" site.

You may not want to wait until the next publication cycle for a changed web page to appear on the site, especially if you corrected errors on a page. To accommodate for this, Oracle Site Studio Publisher has a Publish Now feature, which enables it to search for files on the content server that are marked for immediate publication and then publish only those files. This is a much quicker process, since it does not involve deploying the entire site, but only a limited number of files.

The Publish Now selection is in the Oracle Site Studio Contributor toolbar. When a contributor changes a page and wants it published quickly, they click the Publish Now selection in the Contributor toolbar. For more information on Publish Now in Oracle Site Studio Contributor, see *Using Oracle Site Studio Contributor*.

Contributor can be controlled regarding whether or not the Publish Now feature is available, by setting the feature in Designer. This is easily handled in the menu bar, first selecting **File**, then **Site**, then **Advanced**, and finally **Enable Publish Now** (for more information, see [Section 5.9.1.1, "Site Submenu"](#)). If Enable Publish Now is checked, it is enabled, and contributors working on the web site can mark the individual pages with Publish Now.

Customizing Designer

In Oracle Site Studio Designer, you can change the location and placement of each window and pane and therefore change your working environment.

The Oracle Site Studio Designer interface provides a powerful working environment for creating and managing web sites. When you first open Oracle Site Studio Designer, you see five main areas to work in: the Site Hierarchy pane and the on-screen Properties pane on the left, the workspace and Site Assets pane in the middle, and the Toolbox on the right (see [Section 5.3, "Main Designer Window"](#)).

The arrangement of the workspace and each pane is only a default. Like most applications, you can completely customize the interface if you like. You can, for example, hide, move, and resize panes; modify the available menus and toolbars; and add or modify keyboard shortcuts.

This section covers the following topics:

- [Section 6.1, "Customizing the Panes in Designer"](#)
- [Section 6.2, "Using the Customize Dialog"](#)
- [Section 6.3, "Creating a New Menu"](#)
- [Section 6.4, "Customizing Toolbars"](#)
- [Section 6.5, "Adding Shortcuts to the Tools Menu"](#)
- [Section 6.6, "Assigning Keyboard Shortcuts"](#)
- [Section 6.7, "Customizing Menus"](#)
- [Section 6.8, "Setting Display Options"](#)
- [Section 6.9, "Formatting the Code in Source View"](#)
- [Section 6.10, "Showing or Hiding Warning Messages"](#)
- [Section 6.11, "Setting Miscellaneous Options"](#)
- [Section 6.12, "Changing Log File Settings"](#)

6.1 Customizing the Panes in Designer

In Designer, you can change the location and placement of each window and pane and therefore change your working environment. You can, for example, do the following:

- [Section 6.1.1, "Showing or Hiding Panes"](#)
- [Section 6.1.2, "Resizing a Pane"](#)
- [Section 6.1.3, "Moving or Docking a Pane to a Different Border"](#)

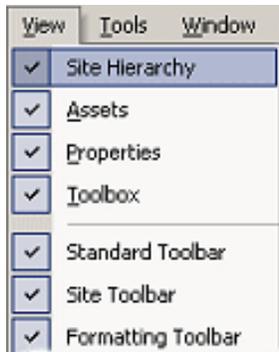
6.1.1 Showing or Hiding Panes

To show or hide panes and toolbars in Designer, perform these tasks:

1. Click **View**.
2. Click the item to show or hide in the interface.

You can show or hide the Site Hierarchy pane, Properties pane, Site Assets pane, Toolbox, Status Bar, Standard Toolbar, Site Toolbar, Formatting Toolbar, and Table Toolbar.

Figure 6–1 *Showing or Hiding Interface Items*



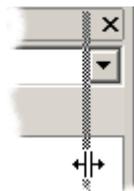
3. The window or toolbar that you selected appears or disappears in the Designer interface.

6.1.2 Resizing a Pane

To resize a pane in Designer, perform these tasks:

1. Place your mouse over the border of a pane.
2. Click and drag the border to resize the pane.

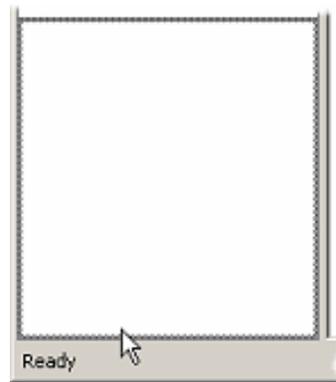
Figure 6–2 *Dragging Pane Borders*



6.1.3 Moving or Docking a Pane to a Different Border

By default, the Site Hierarchy pane and Properties pane are docked on the left, and the Toolbox is docked on the right. You can move these panes around and even dock them to a different border in Designer.

1. Select the title bar of the pane to move.
2. Click and drag the pane to a different location and release the mouse.
3. To dock the window on a different border, click and drag the pane to the new border until you see a gray box surround the area and then release the mouse.

Figure 6-3 Docking a Pane

6.2 Using the Customize Dialog

In addition to changing the size and location of each pane, you can customize the Designer interface. In the Customize dialog, you can customize keyboard commands, right-click menus, shortcuts to outside applications, and much more. Perform the following steps to open the Customize dialog:

1. Click **View**, then select **Customize**.
The Customize dialog opens (see [Section A.1, "Customize Dialog"](#)).
2. Make the necessary changes and then click **Close**.

Dialog Tabs

The Customize dialog includes several tabs:

- **Commands:** Use this tab to add a menu to the Menu bar and also to add commands to menus and toolbars. For more information, see [Section 6.3, "Creating a New Menu."](#)
- **Toolbars:** Use this tab to customize toolbars. For more information, see [Section 6.4, "Customizing Toolbars."](#)
- **Tools:** Use this tab to add shortcuts to outside applications. The shortcut displays in the Tools menu. For more information, see [Section 6.5, "Adding Shortcuts to the Tools Menu."](#)
- **Keyboard:** Use this tab to assign keyboard shortcuts to commands in Designer. For more information, see [Section 6.6, "Assigning Keyboard Shortcuts."](#)
- **Menu:** Use this tab to customize menus in Designer. For more information, see [Section 6.7, "Customizing Menus."](#)
- **Options:** Use this tab to display screentips, large icons in the toolbar, and personalized menus. For more information, see [Section 6.8, "Setting Display Options."](#)
- **Source View:** Use this tab to control the font, size, and other formatting characteristics of the text that opens in the Source View of a page template. For more information, see [Section 6.9, "Formatting the Code in Source View."](#)
- **Warning Dialogs:** Use this tab to show or hide the warnings that appear while working in Designer. For more information, see [Section 6.10, "Showing or Hiding Warning Messages."](#)

- **Miscellaneous:** Use this tab to specify certain, miscellaneous options when Designer starts. For more information, see [Section 6.11, "Setting Miscellaneous Options."](#)
- **Log File:** Use this tab to specify logging options. For more information, see [Section 6.12, "Changing Log File Settings."](#)

6.3 Creating a New Menu

On the Commands tab of the Customize dialog (see [Section A.1.1, "Customize Dialog: Commands Tab"](#)), you can view the menus and commands available on the menu bar in Designer. You can create a menu and populate it with commands from the menus appearing on this tab. In conjunction with options on the Toolbars tab, you can create a toolbar to display frequently used commands together (again by copying commands from this tab).

To create a menu, perform these tasks:

1. Click **View**, then select **Customize**.
2. On the Commands tab, under Categories, click **New Menu**.
3. Under Commands, click **New Menu** and hold down the mouse button.
A small icon appears.
4. Drag the icon to the Menu bar. Release the mouse button.
The menu "New Menu" appears on the Menu bar.
5. Click **New Menu** on the Menu bar, and you see an empty space below the menu name.
You can populate the menu by dragging and dropping commands from other menus into this space.
6. Click **Close** to close the Customize dialog.

Notes

- You can change the name of a menu you create by right-clicking the menu and clicking **Button Appearance** on the popup menu.
- For information on creating a toolbar on the Toolbars tab, see [Section 6.4, "Customizing Toolbars."](#) After you've done that, you can drag and drop commands from the Commands tab onto this toolbar. (When you use drag and drop, you copy commands to the toolbar. The commands remain available in their original menu.)
- For information on customizing the right-click popup menus in Designer, see [Section 6.7, "Customizing Menus."](#)
- To remove a menu that you created, click the menu and, with the mouse button held down, drag it from the menu bar and release the mouse anywhere.

6.4 Customizing Toolbars

On the Toolbars tab of the Customize dialog (see [Section A.1.2, "Customize Dialog: Toolbars Tab"](#)), you can set options to show and hide the toolbars available in Designer. You can also create a toolbar and populate it with commands from the Commands tab dialog; delete any toolbars you may have created; and restore any icons you may have previously removed from the original toolbars.

You can do any of the following:

- [Section 6.4.1, "Showing and Hiding Toolbars"](#)
- [Section 6.4.2, "Creating a New Toolbar"](#)
- [Section 6.4.3, "Resetting a Toolbar"](#)

6.4.1 Showing and Hiding Toolbars

To show and hide toolbars, perform these tasks:

1. Click **View**, then select **Customize**.
2. Click the **Toolbars** tab.
 - To show a toolbar, select the appropriate checkbox.
 - To hide a toolbar, clear the appropriate checkbox.
3. Click **Close** to close the Customize dialog.

You can also show and hide toolbars directly from the View menu by clicking the name of the appropriate toolbar on the menu drop-down list. If you want to display explanatory text with toolbar icons, click **Show text labels**.

6.4.2 Creating a New Toolbar

To create a toolbar, perform these tasks:

1. On the Toolbars tab, click **New**.
2. In the Toolbar Name dialog, type the name of the toolbar and click **OK**.

The toolbar name appears in the Toolbars column. A toolbar icon appears opposite the name.

3. Drag the toolbar icon from the Toolbars tab to the desired position in relation to the other toolbars in Designer. After you release your mouse, the toolbar icon changes, signifying an empty toolbar.
4. To populate the toolbar, open the Commands tab, identify the individual commands you want to display on the toolbar, and drag each icon, in turn, from the Commands box to the toolbar. (You can also drag and drop other toolbar commands.) After you release the mouse, the icon or text for each command appears on the toolbar. (You have simply copied the icon, which remains associated with the original menu.)
5. Click **Close** to close the Customize dialog.

After you create a toolbar, you can customize it further by right-clicking the icon or text and clicking **Button Appearance** on the popup menu.

You cannot delete any of the original toolbars provided with Designer. To delete a toolbar you have created, select the toolbar on the Toolbars tab and click the **Delete** button.

6.4.3 Resetting a Toolbar

If you have removed icons from an original toolbar provided with Designer, you can restore those icons to the toolbar at any time.

To reset a toolbar, perform these tasks:

1. On the Toolbars tab, select the toolbar to restore and click **Reset**.

The toolbar now displays all of the icons originally associated with the toolbar.

2. To restore all of the toolbars, click **Reset All**.
3. Click **Close** to close the Customize dialog.

Note: The Reset and Reset All commands apply only to the default set of toolbars provided with Designer. If you create a toolbar and then remove an icon from that toolbar, the only way to restore the icon is by going to the Commands tab again and dragging that icon from the Commands box then dropping it onto the toolbar.

6.5 Adding Shortcuts to the Tools Menu

On the Tools tab of the Customize dialog (see [Section A.1.3, "Customize Dialog: Tools Tab"](#)), you can add shortcuts to outside applications to the Tools menu in Designer to give you immediate access to these applications.

To add a shortcut to a tool, complete the following:

1. Click **View**, then select **Customize**.
2. Click the **Tools** tab.
3. Click **New** ([Figure 6-4](#)) on the icon bar above the **Menu Contents** pane.

Figure 6-4 *New Shortcut Icon*



A placeholder for the tool appears in the pane.

Insert your cursor in the **Command** text field and type the full path for the executable file of the application you want to be available from the Tools menu.

or

With your cursor in the Command text field, click the **Open** ([Figure 6-5](#)) button opposite the field. In the Open dialog, locate the executable for the application you want to add. Select the file and click **Open**.

Figure 6-5 *Open Icon*



Instead of typing the full path for the executable in the Command line, you may type the file name of the executable only and type the directory for the executable in the Initial Directory field.

4. You can use the Arguments field to direct the application to open a certain file. The argument requires the full path and file name for the executable followed by the full path and file name for the document you want to open. In this instance, the path for the executable should have forward slashes, there should be a space between the two paths, and the path of the document should be enclosed in quotation marks. For example:

```
C:/Program Files/Office/winword.exe "C:\Documents\sa.doc"
```

5. In the Initial Directory field, type the directory for the executable only if the executable file name by itself is entered in the Command field (step 4). Otherwise, leave it empty.
6. Click **Close** to close the Customize dialog.

The shortcut for the application is now available on the Tools menu. If you have entered an argument, not only does the application open when you select it from the Tools menu, but the document specified in the argument opens, too.

You can delete a tool at any time by selecting the tool in the Menu Contents pane and clicking Delete (Figure 6-6).

Figure 6-6 Delete Icon



You can change the order in which tools are displayed in the Tools menu by clicking the **Move Up** or **Move Down** icon (Figure 6-7 and Figure 6-8, respectively).

Figure 6-7 Move Up Icon



Figure 6-8 Move Down Icon



6.6 Assigning Keyboard Shortcuts

On the Keyboard tab of the Customize dialog (see [Section A.1.4, "Customize Dialog: Keyboard Tab"](#)), you can assign keyboard shortcuts to commands in the menus of Designer. Many of the commands have one or more keyboard shortcuts assigned by default. You can change or delete these keyboard shortcuts, and you can add new ones.

To assign a keyboard shortcut, perform these tasks:

1. Click **View**, then select **Customize**.
2. Click the **Keyboard** tab.
3. Under **Category**, select the menu that contains the command to assign a shortcut to.
4. Under **Commands**, select the command to assign the shortcut to.

A description for the command appears in the Description panel. Any keyboard shortcuts already assigned appear under Current Keys.

5. From the Set Accelerator For drop-down list, select **Default** if you want the keyboard shortcut to be available generally when you are working in Designer.
6. Place your cursor in the **Press New Shortcut Key** text box. Then, on your keyboard, press the actual key or combination of keys you want to assign for that shortcut.

To use the Shift, Ctrl, or Alt key with another letter, press **Shift**, **Ctrl**, or **Alt** followed by the letter.

7. Click the **Assign** button.
8. Click **Close** to close the Customize dialog.

6.7 Customizing Menus

On the Menu tab of the Customize dialog (see [Section A.1.5, "Customize Dialog: Menu Tab"](#)), you can customize the behavior of popup menus in Designer.

You can do any of the following:

- [Section 6.7.1, "Adding a Command to a Popup Menu"](#)
- [Section 6.7.2, "Deleting a Command From a Popup Menu"](#)
- [Section 6.7.3, "Controlling the Display of Menus"](#)

6.7.1 Adding a Command to a Popup Menu

To add a command to a popup menu, perform these tasks:

1. Click **View**, then select **Customize**.
2. Click the **Menu** tab.
3. Under Select Context Menu, select the popup menu you want to customize.
The popup menu appears to the upper-left of the Customize dialog.
4. In the Customize dialog, click the **Commands** tab.
5. Under Categories, select the appropriate menu. Under Commands, select the command you want to add to the popup menu. Drag the command onto the popup menu and position it where you want it to appear on the menu. Let go of the mouse.
6. Click **Close** to close the Customize dialog.

6.7.2 Deleting a Command From a Popup Menu

To delete a command from a popup menu, perform these tasks:

1. Click **View**, then select **Customize**.
2. Click the **Menu** tab.
3. Under **Select Context Menu**, select the popup menu you want to edit.
The popup menu appears to the upper-left of the Customize dialog.
4. On the popup menu, right-click the command you want to delete and click **Delete** on the popup menu that appears.
5. Click **Close** to close the Customize dialog.

You can restore any deleted command to a popup menu by selecting the menu from the Select Context Menu drop-down list and clicking the **Reset** button.

6.7.3 Controlling the Display of Menus

You can control the on-screen appearance of menus in two ways. You can turn on or off the "shadow" that displays around them, and you can control how they open. These options apply to menus on the Menu bar and to the menus that appear when

you right-click a section or page template in the Site Hierarchy pane or a fragment in the Toolbox.

To control how menus appear on-screen, perform these tasks:

1. Click **View**, then select **Customize**.
2. Click the **Menu** tab.
3. Under **Select Context Menu**, select any popup menu.

Note: Whichever menu you select, your selection in step 4 applies to all menus in Designer.

4. Under **Menu Animations**, select the animation to use:
 - **None:** The list of menu commands appears in the usual way.
 - **Unfold:** The list of menu commands opens from top-left to bottom-right.
 - **Slide:** The list of menu commands opens from top to bottom.
 - **Fade:** The list of menu commands slowly appears on-screen.
5. To display a shadow effect around menus, click the **Menu shadows** checkbox. To hide the shadow effect, clear the checkbox.
6. Click **Close** to close the Customize dialog.

For information on customizing the menus on the Menu bar in Designer, see [Section 6.3, "Creating a New Menu."](#)

6.8 Setting Display Options

On the Options tab of the Customize dialog (see [Section A.1.6, "Customize Dialog: Options Tab"](#)), you can control display settings, like screentips, large icons on the toolbar, and personalized menus.

To set display options, perform these tasks:

1. Click **View**, then select **Customize**.
2. Click the **Options** tab.
3. Under **Toolbar**, click **Show ScreenTips on toolbars** if you want the popup text to identify an icon as you pass your cursor over it. If you want the screentips to also display the shortcut key assigned to the command, click **Show shortcut keys in ScreenTips**. To display large icons, click **Large Icons**.
4. Under **Personalized Menus and Toolbars**, select the **Menus show recently used commands first** checkbox if you want the menu commands to be ordered in terms of their frequency of use.

If you do so, you may also select **Show full menus after a short delay**.

5. Click **Close** to close the Customize dialog.

You can click the **Reset my usage data** button at any time to return Designer to its original state.

6.9 Formatting the Code in Source View

On the Source View tab of the Customize dialog (see [Section A.1.7, "Customize Dialog: Source View Tab"](#)), you can change the way code on a page template appears when viewed in Source view. You can change the font face and font size and also apply automatic formatting options.

In this pane, you also control which DTD the page should conform to in the DOCTYPE menu.

To format the code that opens in Source view, perform these tasks:

1. Click **View**, then select **Customize**.
2. Click the **Source View** tab.
3. Under Editor, from the Font drop-down list, select the font from those available, and from the Size drop-down list, select the font size you want the code to display in.
4. Under Automatic Formatting, select **Apply automatic formatting when switching to Source view** to automatically format the code according to the options on this tab.

For a description of each formatting option, see [Section A.1.7, "Customize Dialog: Source View Tab."](#)

5. Click **Close** to close the Customize dialog.

6.10 Showing or Hiding Warning Messages

On the Warning Dialogs tab of the Customize dialog (see [Section A.1.8, "Customize Dialog: Warning Dialogs Tab"](#)), you can control which warning messages display and which ones remain hidden as you work in Designer.

To show and hide warning messages, perform these tasks:

1. Click **View**, then select **Customize**.
2. Click the **Warning Dialogs** tab.
3. Select the checkboxes for the warning messages to display and clear the checkboxes for the warning message to remain hidden.

Click **Check All** if you want all messages to display or **Clear All** if you want all messages to remain hidden.

4. Click **Close** to close the Customize dialog.

Note: By default, certain warning messages display when you first run Designer.

6.11 Setting Miscellaneous Options

On the Miscellaneous tab of the Customize dialog (see [Section A.1.9, "Customize Dialog: Miscellaneous Tab"](#)), you can change various options in Designer, such as whether to connect to the last site you worked on and whether to prevent fragments unrelated to your site from appearing in the Toolbox.

To change the miscellaneous options in Designer, perform these tasks:

1. Click **View**, then select **Customize**.

2. Click the **Miscellaneous** tab.
3. To display only the fragments written in the same language (HCSP/JSP or ASP) as your site in the Toolbox, check **Filter fragments in other languages**.
4. To display only the fragments that belong to your site (and not other sites in the content server) in the Toolbox, check **Filter libraries from other sites**.

Fragments in fragment libraries not identified as part of your site no longer appear in the Toolbox. To make a fragment part of your site, add it as a site asset to the "Fragment Libraries" category.
5. To display only the fragments that you have created (and not the default ones that ship with Oracle Site Studio) in the Toolbox, check **Filter default libraries**.
6. To automatically connect to the last site you worked on when starting Designer, check **Reconnect at startup**.
7. To display the **Project Status** icon in Designer's status bar, check **Enable project status checking**. For more information, see [Section 7.3.2, "Viewing the Project Status in Designer."](#)
8. Click **Close** to close the Customize dialog.

6.12 Changing Log File Settings

On the Log File tab of the Customize dialog (see [Section A.1.10, "Customize Dialog: Log File Tab"](#)), you can specify when a log file is created and how detailed it should be.

To change the log file settings, perform these tasks:

1. Click **View**, then select **Customize**.
2. Click the **Log File** tab.
3. To create a log file each time you open Designer, check the box **Reset log file each time Designer starts**.
4. To create a detailed log of all communication with the content server, check the box **Log details of all communication with the Content Server**.
5. Click **Close** to close the Customize dialog.

Creating Your Web Site

The first thing you do in Oracle Site Studio is connect to an Oracle Content Server and create your assets. You do this using the Connection Manager in Oracle Site Studio Designer. With the Connection Manager, you identify a content server, create a web site on that server (more specifically, a project file), and create a connection name that is used by Oracle Site Studio Designer to connect to the site.

As soon as you've done this, you can start constructing your site assets into a web site, using a site hierarchy to define the arrangement of the information. A site hierarchy comprises the individual sections of your site, like *Products*, *Services*, and *About Us*. In each section, there is a primary page and often a secondary page.

You can spend a considerable amount of time setting up the site and site hierarchy, which is discussed next.

- [Section 7.1, "Site Connections"](#)
- [Section 7.2, "Site Addressing"](#)
- [Section 7.3, "Project Files"](#)
- [Section 7.4, "Starting and Stopping a Web Site"](#)

7.1 Site Connections

With Oracle Site Studio, you create your site on the content server and then associate various files (page templates, fragments, data files, and so on) with the site. To create the site, you must create what is called a site connection in Designer.

Designer uses a site connection to connect to and update web sites stored in the content server. A site connection works much like a typical shortcut that you might add to your desktop for quick access to another program or file. You can add, edit, and delete a site connection without actually affecting the web site in the content server. You can add as many site connections as you like, each one going to a different web site.

Note: Oracle Site Studio 11gR1 is fully backward compatible. This means that you can use Oracle Site Studio Designer 11gR1 to work with sites created in earlier Oracle Site Studio releases. It is important to note, though, that these sites continue to work in "legacy" mode; that is, they use the pre-10gR4 architecture and they do not take advantage of the architecture and features introduced in Oracle Site Studio 10gR4.

Perform the steps outlined in these topics to manage a site connection:

- [Section 7.1.1, "Creating a New Site Connection"](#)
- [Section 7.1.2, "Editing a Site Connection"](#)
- [Section 7.1.3, "Deleting a Site Connection"](#)

7.1.1 Creating a New Site Connection

To create a web site, you must create a site connection that is used to connect to, and update, the web site on the content server. If this is the first time you have opened Designer, you are automatically taken to this dialog.

To create a site connection, perform these tasks:

1. Click the **Site Connection Manager** icon in the Site toolbar (or select **File**, then **Site**, and then **Connection Manager**) to open the Site Connection Manager dialog. For more information, see [Section A.2, "Site Connection Manager Dialog."](#)

Figure 7–1 Site Connection Manager icon



2. Click **New** to open the Site Connection Details dialog. For more information, see [Section A.3, "Site Connection Details Dialog."](#)
3. In the **Server Cgi URL** text box, enter the Cgi URL address of the content server that hosts your site.

Rather than type the complete address of your server, you can choose an address in the menu (replacing *<server>* with the name of your server).

The default address, `http://<server>/stellent/idcplg`, is for Oracle Content Server versions 7.x, 10gR3, and 10gR4. The two other addresses available from the drop-down list are for legacy versions: `http://<server>/intradoc-cgi/idc_cgi_isapi.dll` for Windows, and `http://<server>/intradoc-cgi/nph-idc_cgi.exe` for UNIX.

4. Click **Connect**. You may be prompted to enter your login credentials.
5. Click **New** to create a site on the content server.

The Create New Site dialog appears (see [Section A.4, "Create New Site Dialog"](#)).

Or, to create a site connection using an existing web site on the server, choose it from the list of names beside "Site Label," and then skip to step 10.

6. Enter an **ID** for the web site. (The ID is used throughout your site, for example, in the web site URL and in the metadata for the files referenced by the site.)
7. Enter a **Name** for the web site.

Since the Site ID field is a full-text index field, you should take care to not use a Site ID that would appear in the full-text search engine's "stop list." If you do, your Oracle Site Studio site will not work as expected in Oracle Site Studio Designer. An example of this would be a web site for the World Health Organization, which might use the shortened site ID "who."

The name appears in Designer and the "Web Sites" menu in the content server. It can be a friendly name with spaces, if you like.

Important: The Site ID should not contain the underscore character "_" as this may cause problems with indexing the web sites and full-text metadata fields. The underscore is, by default, used as a word break when indexing, and a wild card while searching.

8. Choose **hcsp/jsp** as the type. (The site type governs other scripting choices in Designer, in particular, the scripting format for page templates and fragments.)

Important: You can create a site using JSP or ASP. However, only sites created with HCSP can take advantage of the features introduced with 10gR4. In other words, JSP and ASP sites behave as pre-10gR4 sites.

Note: You cannot change the ID or Type after you close this dialog.

9. Click **OK** to return to the Site Connection Details dialog.
10. Oracle Site Studio automatically inserts a connection name, which you see when you connect to your site in Designer (the Site Hierarchy menu and the Recent Sites menu). You may use this name or enter a different one.
11. Click **OK** to return to the Site Connection Manager.
12. Check **Automatically connect when done** (if it is not already checked), and then click **Done**.

Designer opens the site from the content server and download the available fragments that are stored there. Your web site opens in the Site Hierarchy pane.

Figure 7-2 Web site in Site Hierarchy page



7.1.2 Editing a Site Connection

You can edit the site connections that you create in Designer. You can change the name of the site connection, the content server associated with the site connection, and the web site associated with the site connection. (You cannot change the site ID, name, or type.)

To edit a site connection, you must first disconnect from the site if the site is currently open in Designer.

To edit a site connection, perform these tasks:

1. If you are currently connected to the site you want to edit, click the **Disconnect** icon (Figure 7-3) in the Site toolbar (or select **File**, then **Site**, and then **Disconnect**).

Figure 7-3 Disconnect icon

2. Click the **Site Connection Manager** icon (Figure 7-4) in the Site toolbar (or select **File**, then **Site**, and then **Connection Manager**) to open the Site Connection Manager dialog. For more information, see [Section A.2, "Site Connection Manager Dialog."](#)

Figure 7-4 Site Connection Manager icon

3. Highlight the connection name and then click **Edit**.
4. In the Site Connection Details dialog, make the necessary changes to the **Server Cgi URL**, the **Site Label**, or the **Connection Name**. For more information, see [Section A.3, "Site Connection Details Dialog."](#)
5. Click **OK** to return to the Site Connection Manager.
6. With the site connection you just edited still highlighted, check **Automatically connect when done** (if it is not already selected), and then click **Done**.

Note: You cannot change the ID or Type after you create a site. These values are used throughout your site.

7.1.3 Deleting a Site Connection

You can remove the site connections that you create in Designer. This removes only the connection and not the actual web site on the content server (see note below). To remove a site connection, you must first disconnect from the site if it is currently open in Designer.

To delete a site connection, perform these tasks:

1. If you are currently connected to the site to edit, click the **Disconnect** icon (Figure 7-5) in the Site toolbar (or select **File**, then **Site**, and then **Disconnect**).

Figure 7-5 Disconnect icon

2. Click the **Site Connection Manager** icon (Figure 7-6) in the Site toolbar (or select **File**, then **Site**, and then **Connection Manager**) to open the Site Connection Manager dialog. For more information, see [Section A.2, "Site Connection Manager Dialog."](#)

Figure 7-6 Site Connection Manager icon

3. Highlight the connection name and click **Delete**.
4. Click **Yes** to confirm the deletion.

5. Click **Done** to close the Site Connection Manager.

To delete the actual web site on the content server, you must log on to the server and delete the project file created by Oracle Site Studio. Before you do this, you must stop the running web site (see [Section 7.4, "Starting and Stopping a Web Site"](#)) so that revisions of the project file are not checked in (automatically) when you are trying to delete it.

7.2 Site Addressing

Like any web site, in Oracle Site Studio you must associate an address with your web site so that you, contributors, and visitors can locate the site with a web browser.

Oracle Site Studio can identify a web site using a folder address, a domain address, or a combination of both. The Administrator can specify a domain, a folder, or a combination to map to a particular web site.

When you first set up your site in Designer, you have a folder name address, which comprises the name of the hosting computer and the site ID. This address works well for internal purposes, such as a departmental web site or a demonstration site. However, if you intend to make the site publicly available on the Internet, you should configure one or more domain names and point them to the site.

A folder-based site, using a folder address, uses only the folder to identify a site. An example of this would be `www.oracle.com/SiteStudio`, where the folder `SiteStudio` identifies the specific site on the domain.

A domain-based site uses simply a domain to identify a site. Examples would be `www.oracle.com`, or `sales.mycompany.com`. The domain identifies the site.

The combination of both uses a domain and a folder (or any number of folders). Common examples of this are to use language folders, such as `www.oracle.com/en/` or `www.oracle.com/fr/`. Using the combination of domain and folder also creates a convenient way to move your site from design to the production system.

A domain address requires additional DNS configuration that the site manager or site administrator must perform.

These topics cover site addressing in greater detail:

- [Section 7.2.1, "Mapping Domain Names to Your Site"](#)
- [Section 7.2.2, "Changing the Path Used in a Site Address"](#)

7.2.1 Mapping Domain Names to Your Site

A domain name is a unique name that identifies a web site. You can point one or more domain names to your Oracle Site Studio web site, and you most likely do this if you plan to make your site publicly available on the Internet.

The way to do this is get the domain name, configure it on a name server, and then map the domain name to your site using Oracle Site Studio. This topic only discusses the last step, mapping the domain name. For help configuring the domain name, contact the appropriate third party.

There are two places in Oracle Site Studio where you can map a domain name:

- [Section 7.2.1.1, "Mapping a Domain Name Using Designer"](#)
- [Section 7.2.1.2, "Mapping a Domain Name Using the Manage Web Site Addresses Page"](#)

7.2.1.1 Mapping a Domain Name Using Designer

To map a domain name using Oracle Site Studio Designer, perform these tasks:

1. Select **File**, then **Site**, then select **Advanced**, and then **Edit Site Addresses**.
This opens the Site Addresses dialog (Figure 7-7). For more information, see Section A.5, "Site Addresses Dialog."
2. To add a domain, click the **New Site Address** icon (Figure 7-7) and type a name in the space provided.

Figure 7-7 *New Site Address icon*



When you enter a domain name, do not add the protocol specifier (http or https). Only enter the domain name.

3. Press **Enter** on your keyboard to add the domain.
Repeat these steps for each additional domain name to add.
4. To make the domain name the default one used by Oracle Site Studio, click **Set Default**.
5. To remove an existing domain name, select it and click the **Delete** icon (Figure 7-8).

Figure 7-8 *Delete icon*



6. Click **OK**.

7.2.1.2 Mapping a Domain Name Using the Manage Web Site Addresses Page

To map a domain name using the Manage Web Site Addresses page, perform these tasks:

1. Log onto the content server as an administrator and click **Administration**.
2. Click **Oracle Site Studio Administration** to open the Oracle Site Studio Administration page.
3. Click **Manage Web Site Addresses** to open the Manage Web Site Addresses page.
4. To add a domain, select the web site in the **Web Site** menu.
5. Enter the domain name in the **Address** field.
6. Click **Add** to add the domain name to the web site.
The page refreshes, showing the web site address in the list of addresses.
7. To make the domain name the default one used by Oracle Site Studio, click the **Actions** icon (Figure 7-9) beside the name and choose **Select As Default Address**.
8. To remove an existing domain name, click the **Actions** icon (Figure 7-9) beside the name and choose **Remove Site Address**.

Figure 7–9 Actions icon

You can map multiple domain names to your site. When you do this, users see whatever domain name they entered the site with the entire time they browse the site. The domain name that you specify as a default is used by Oracle Site Studio when you preview your site, assign content to a region, and so forth.

7.2.2 Changing the Path Used in a Site Address

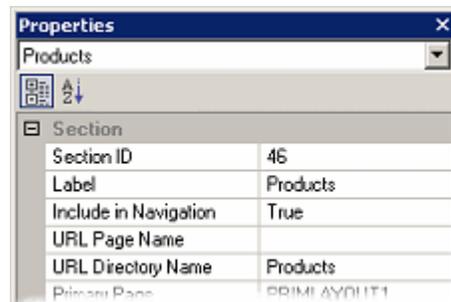
When you open your site using a domain-based address, you see the http protocol (or https, depending on how the content server is configured), followed by the domain name, followed by the section, followed by a web page. For example:

```
http://www.mydomain.com/products/index.htm
```

When you open your site using a folder-based address, you see the http protocol (or https, depending on how the content server is configured), followed by the name of the computer (that hosts the site), followed by the ID of the web site, followed by the section, followed by a page name. For example:

```
http://UserPC/mysite/products/index.htm
```

You can change the name that appears in this path by changing the properties of the site, the properties of a section, and the properties for a page template. You change these values using the Properties pane (Figure 7–10).

Figure 7–10 Properties Pane

To change the path names used in a site address, perform these tasks:

Name in path	How to change its value in the URL
To change the server-relative name in a path-based address	<p>You can override the value of the Site ID by specifying a URL Directory Name for the root of the site hierarchy.</p> <ul style="list-style-type: none"> ■ Open the Site Hierarchy pane. ■ Click the root of the site. ■ Open the Properties pane. ■ Enter a value for URL Directory Name.
To change the name of a section	<ul style="list-style-type: none"> ■ Open the Site Hierarchy pane. ■ Click the section you want to change. ■ Open the Properties pane. ■ Enter a value for URL Directory Name.

Name in path	How to change its value in the URL
To change the name for a primary page	<ul style="list-style-type: none"> ■ Open the Site Hierarchy pane. ■ Click the page template you want to change. ■ Open the Properties pane. ■ Enter a value for URL Page Name.

The name used for secondary pages is the content ID (*dDocName*) of the contributor data file or native document appearing on that page. If desired, you can specify a different value. For details, see *Technical Reference for Oracle Site Studio*. When you specify a URL Directory Name, you can change the label of a section as many times as you like without affecting the URL.

7.3 Project Files

When you create a web site, Oracle Site Studio creates a project file and checks it into the content server. The project file stores your site hierarchy and key information about your web site. Whenever you open and edit your site hierarchy in designer, you are editing the project file.

You do not edit the project file directly, but there are occasions when you perform steps that affect the project file:

- [Section 7.3.1, "Specifying the Metadata That is Assigned to a Project File"](#)
- [Section 7.3.2, "Viewing the Project Status in Designer"](#)
- [Section 7.3.3, "Committing Changes to a Project File"](#)
- [Section 7.3.4, "Viewing the Content Information Page for the Project File"](#)

7.3.1 Specifying the Metadata That is Assigned to a Project File

When you create a web site for the first time (see [Section 7.1.1, "Creating a New Site Connection"](#)), Oracle Site Studio creates a project file and checks it into the content server. Like any other content item, there are metadata values that must be assigned to this file.

You can specify the metadata that is assigned to all project files using the "Set Project Default Document Information" page in the content server.

To specify the metadata for new project files, perform these tasks:

1. Log onto the content server as an administrator and click **Administration**.
2. Click **Oracle Site Studio Administration** to open the Oracle Site Studio Administration page.
3. Click **Set Project Default Document Information** to open the Set Project Default Document Information page.
4. Select the metadata that you would like to use for your Oracle Site Studio project files.
5. Click **Update**.

7.3.2 Viewing the Project Status in Designer

As you work in Designer, you may notice a project status icon in the lower right corner. This icon tells you if you are connected to the project file on the content server and if your site hierarchy and the project file are synchronized with one another.

Here is a summary of each icon you may see in the project status area in Designer:

Icon	Definition
	Your site hierarchy matches the project file on the content server.
	Your site hierarchy does not match the project file on the content server. You should refresh your site hierarchy.
	The up-to-date status cannot be determined. Either the content server is down, or you have lost your network connection. To resolve this, troubleshoot your network connection or the system running the content server.
	Designer is not connected to a web site. To resolve this, reconnect to a web site (see Section 7.1, "Site Connections").

7.3.3 Committing Changes to a Project File

The changes you make to your site hierarchy in Designer affect the contents of the project file on the content server. Oracle Site Studio automatically checks in a revision of the project file every ten minutes. To commit changes to the project file sooner than this scheduled time, follow these steps.

To commit site hierarchy changes to the project file, perform these tasks:

1. In Designer, select **File** then **Site**, then select **Advanced** and then **Commit Project Changes**.
2. Click **OK** to the confirmation message.

Note: Because of the way Oracle Site Studio manages project files, there is no worry about losing your site hierarchy changes; even if your connection with the content server is disrupted or the server is temporarily down.

7.3.4 Viewing the Content Information Page for the Project File

You can view your project file's content information page at any time from Designer. Viewing the content information page can be useful when you want to see the details of your project file in the content server, update its metadata, download the file, and so on.

To view the content information page for the project file, select **File** then **Site**, then select **Advanced** and then **View Project DOC_INFO** in Designer.

This takes you to the Content Information page for the project file in the content server.

7.4 Starting and Stopping a Web Site

You can start and stop a web site in Oracle Site Studio, similar to the way you start and stop web sites on a web server. You might do this, for example, to troubleshoot the site, to restart the site after a global update, or prevent content from being viewed or downloaded.

Before you stop the site, you should create a temporary page that opens instead of the site when the site is stopped. This way, users know why the site is not accessible.

You can do either of the following:

- [Section 7.4.1, "Stopping the Web Site"](#)
- [Section 7.4.2, "Configuring a Temporary Page for the Stopped Site"](#)

7.4.1 Stopping the Web Site

To stop the web site, perform these tasks:

1. Open the Oracle Site Studio Administration page.
2. Click **Manage Web Sites** to open the Manage Web Sites page.
3. Highlight a web site and then click **Stop Web Site**.
4. When you are finished performing the necessary administrative tasks on your site, return to the Manage Web Sites page, highlight your site, and click **Start Web Site**.

7.4.2 Configuring a Temporary Page for the Stopped Site

To configure a temporary page for when the site is stopped, perform these tasks:

1. Create a web page stating that your site is temporarily down and check the file into the content server.
2. In Designer, highlight your site or a section in the Site Hierarchy pane.
3. Open the Properties pane.
4. Beside **Stopped Site Page**, click the **Additional Information** icon ([Figure 7-11](#)) and browse to the web page to use when the site is stopped.

Figure 7-11 *Additional Information Icon*



Working With Templates

Oracle Site Studio uses templates to place and arrange content on a web site. Templates are HTML files; page templates are complete files with <HEAD> and <BODY> tags, while subtemplates and region templates are partial HTML sections, placed within a page template between the <BODY> tags, and are used only for content related to the body.

This section covers the following topics:

- [Section 8.1, "About Templates"](#)
- [Section 8.2, "Managing Templates as Site Assets"](#)
- [Section 8.3, "Understanding the Contribution Regions"](#)
- [Section 8.4, "Inserting Placeholders"](#)
- [Section 8.5, "Inserting Objects"](#)
- [Section 8.6, "Inserting Fragments"](#)
- [Section 8.7, "Working With Text"](#)
- [Section 8.8, "Arranging Objects on the Template"](#)
- [Section 8.9, "Applying CSS Classes"](#)
- [Section 8.10, "Working With HTML Tables"](#)
- [Section 8.11, "Viewing Templates in Designer"](#)
- [Section 8.12, "Region Templates and Region Definitions"](#)
- [Section 8.13, "Working With Native Documents and Conversion Definitions"](#)

8.1 About Templates

Templates are the files used to place and arrange content on a web site. Templates are HTML files; page templates are complete files with <HEAD> and <BODY> tags, while subtemplates and region templates are partial HTML sections, placed within a page template between the <BODY> tags, and are used only for content related to the body.

The way Oracle Site Studio handles web sites is unlike typical web site development tools. One area where this is clearly seen is in the use of page templates, subtemplates, and region templates.

You can add whatever static content and dynamic content you like to any one of these templates. However, as you spend more time with each of these templates, you see why it makes more sense to add relatively little static content directly to the page.

Adding content directly to the page limits reusability and makes localization more difficult, among other things.

The content that you add directly to a template is likely to be information that remains static, or consistent, across the site. It is what establishes the basic look and feel of the site, and may include such things as background colors, background images, and HTML tables (to position text and graphics).

The rest of the content is built from the contributor data files, which are updated by contributors and are associated with the templates. The templates control how the data is displayed, and the definitions control which data is displayed.

The templates are each similarly used to combine the static and dynamic content to complete each part of the site they control.

This section is about the three types of templates as viewed in the Site Assets pane:

- [Section 8.1.1, "About Page Templates"](#)
- [Section 8.1.2, "About Region Templates"](#)
- [Section 8.1.3, "About Subtemplates"](#)
- [Section 8.1.4, "About Cascading Style Sheets"](#)

8.1.1 About Page Templates

Page templates are fully-formed HTML files that define the layout and complete structure of a web page, 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). They provide the framework within which site content is displayed. The number of page templates required for a web site depends on the site complexity, but usually a small number can easily be used.

When fewer page templates are used, site-wide changes are made much more easily. The page template represents the asset that is a complete web page in HCSP, JSP, or ASP. Just as on a region template, or a subtemplate, you can place anything directly on the template, like text, or an image, or a fragment, and it appears on every instance that template is used. This is also true of subtemplates and region templates.

Page templates consist of standard HTML layout and formatting code, along with Oracle Site Studio tags to specify where fragments and/or placeholders go. As such, they are typically light-weight in that they only contain high-level references to where contribution regions go on the page; they do not specify anything about what goes in these regions, both in terms of content and visual presentation. That is handled by region definitions and templates. Page templates typically include site-wide graphics, such as corporate banners or page layout images, and other recurring, non-editable content, such as navigation aids.

As you reuse the page template, the objects arranged on it are displayed each time the page template opens. Also, when you use the page templates efficiently, then you can create an entire web site with few (or even one) page templates, ensuring that the standard look and feel of your web site is maintained while also making the site management as simple as possible.

For more information, see [Section 3.12, "Page Templates."](#)

8.1.2 About Region Templates

Region templates are partial HTML files that define the layout and look-and-feel of the data in contribution regions within web pages. They are partial HTML files in that

they do not have a head and body section. This allows them to be inserted into other HTML code as the web pages are generated for the Oracle Site Studio site.

These templates consist of standard HTML layout and formatting code, along with Oracle Site Studio tags to specify where elements (from contributor data files) or dynamic conversions (of native documents) are placed. Some elements from contributor data files may be displayed in some region templates, but not in others, which allows the information to be reused across different pages. Region templates are used to display data within the placeholder and are used to arrange the elements (limited by the region definition) into the layout desired in your web site. For more information, see [Section 3.10, "Region Templates and Region Definitions."](#)

Since the creation of a region template requires you associate it with a region definition, it is recommended that you create region definitions first, then create region templates.

8.1.3 About Subtemplates

Subtemplates are the same as page templates, but with one important difference: subtemplates do not have <HTML>, <HEAD>, and <BODY> sections. As such, they are essentially chunks of HTML code that can be inserted in page templates. Subtemplates may contain very simple HTML code, but they can also be quite complex. The code in a subtemplate is treated exactly as it would be when placed directly in a page template, whether it is a script or a CSS file or any other inserted asset.

Subtemplates can only be placed within placeholders, and subtemplates may contain their own placeholder or placeholders. Subtemplates are often used as a way to divide one placeholder, singly placed on a page template, into multiple placeholders. This helps make a page template that is more easily reused.

8.1.4 About Cascading Style Sheets

Cascading style sheets (CSS) are used to control positioning, layout, and other styles on a page. Stylesheets are edited in source view. They are usually added to other objects such as a page template or a subtemplate, just as it would be inserted in any static HTML page.

8.2 Managing Templates as Site Assets

All templates, just as all other site assets, are easily managed through the Site Assets pane. Here you can add new templates, modify the information on existing templates, and remove templates from the site.

This section covers the following topics:

- [Section 8.2.1, "Creating a New Template"](#)
- [Section 8.2.2, "Creating a Page Template From the Site Hierarchy"](#)
- [Section 8.2.3, "Copying a Template"](#)
- [Section 8.2.4, "Viewing the Content Information of a Template"](#)
- [Section 8.2.5, "Viewing the Content Information of a Page Template from the Site Hierarchy"](#)
- [Section 8.2.6, "Adding a Template to a Site"](#)
- [Section 8.2.7, "Removing a Template from a Site"](#)

8.2.1 Creating a New Template

To create a template, perform these tasks:

1. From the list of asset types, select the template type to create.
2. Click the **Create New** icon (Figure 8–1), select **New**, and then select the template type (this will most always be HCSP).

The Assign Info Form opens for you to check the template into the content server.

Figure 8–1 Create New Icon



3. Enter appropriate values for the Assign Info Form.
For guidelines on naming Oracle Site Studio assets, see [Section 4.2, "Naming Site Assets."](#)
4. When complete, click **Assign Info**.
5. The template is created.

There is an additional way to create page templates after the site hierarchy is created. For more information, see [Section 8.2.2, "Creating a Page Template From the Site Hierarchy."](#)

8.2.2 Creating a Page Template From the Site Hierarchy

Another method of creating page templates is available, but only after you have created a site hierarchy:

1. In the site hierarchy, right-click a section and then select either **Select Home Page**, **Select Primary Page**, or **Select Secondary Page**, depending on where the page template is used.

Depending on which you selected from the menu, one of these dialogs opens:

- [Section A.11, "Select \(Home\) Page Template Dialog"](#)
- [Section A.12, "Select \(Primary\) Page Template Dialog"](#)
- [Section A.13, "Select \(Secondary\) Page Template Dialog"](#)

2. Select **Create a new page template**.
3. Click **OK**.
4. Add the information to check the page template into the content server.

For guidelines on naming Oracle Site Studio assets, see [Section 4.2, "Naming Site Assets."](#)

5. The page template is created and associated with the section.

When you create a page template this way, the contextual data is presented in the Preview tab. If you create a page template through the site assets pane, then you must assign the page template to a section to see the data which opens based on which section the page is in.

8.2.3 Copying a Template

You can select a template from the list to copy, select a template from the content server to copy, or select a template from your local instance to copy.

Copying a template from the Site Assets pane

1. In the menu on the Site Assets pane, select the template type.
2. Select the template you want to copy from the list.
3. Click the **Create New** icon (Figure 8–2), select **Copy**, and then **Selected**:
The Assign Info Form opens for you to check the item into the content server.

Figure 8–2 Create New Icon



4. Enter appropriate values for the Assign Info Form.
5. When complete, click **Assign Info**.
6. The template is copied.

Copying a template from the content server

1. In the menu on the Site Assets pane, select the template type.
2. Click the **Create New** icon (Figure 8–2), select **Copy**, and then **from Server**.
A search results page opens.
3. Click the corresponding **Select** button of the one you want to copy.
The Assign Info Form opens for you to check the item into the content server.
4. Enter appropriate values for the Assign Info Form.
5. When complete, click **Assign Info**.
6. The template is copied.

Copying a template from your local instance

1. In the menu on the Site Assets pane, select the template type.
2. Click the **Create New** icon (Figure 8–2), select **Copy**, and then **from Local**.
An open file dialog opens.
3. On your local instance, navigate to the template you want to copy.
4. Select the file and click **Open**.
The Assign Info Form opens for you to check the item into the content server.
5. Enter appropriate values for the Assign Info Form.
6. When complete, click **Assign Info**.
7. The template is copied.

8.2.4 Viewing the Content Information of a Template

To view content information for a template, perform these tasks:

1. In the menu on the Site Assets pane, select the template type.

2. Select a template from the list.
3. Click the **Doc Info** icon (Figure 8–3).
The content information page opens.

Figure 8–3 Doc Info icon



8.2.5 Viewing the Content Information of a Page Template from the Site Hierarchy

There may be times when you want to view the content information page for a page template. You can use the features available in the content server to update the metadata, send a link to the file by e-mail, subscribe to it, and so on.

You can view the content information page using the site hierarchy pane.

To view the metadata of a page template, right-click the desired page template in the site hierarchy pane and choose **View Document Info**.

8.2.6 Adding a Template to a Site

To add a template to a site, perform these tasks:

1. In the menu on the Site Assets pane, select the template type.
2. Click the **Add to Site** icon (Figure 8–4).
A search results page opens.

Figure 8–4 Add to Site Icon



3. Select the templates to add, click **Oracle Site Studio**, and then **Select Marked Documents**.
4. Depending on your configuration, you may receive a caution that you are about to add existing asset(s) to your site. Click **OK**.

The templates are now associated with the web site.

8.2.7 Removing a Template from a Site

To remove a template from a site, perform these tasks:

1. In the menu on the Site Assets pane, select the template type.
2. From the list, select the template to be removed and click the **Remove From Site** icon (Figure 8–5).

When you remove a template, you are simply removing its association with a site. You are not deleting the file from the content server.

Figure 8–5 Remove from Site Icon



8.3 Understanding the Contribution Regions

The area that the contributor can edit is called the contribution region. This area is defined by the designer by using a placeholder. Within that placeholder, the subtemplate (if used) and region template are used to control how the data associated within the contribution region is displayed. The definitions - placeholder, region, and element, are used to control how the contributor edits and interacts with that data.

Element definitions are used to determine what toolbar icons and editing capabilities are available to the contributor. The elements are arranged on the region template. The region template is the smallest "chunk" of HTML that is controlled as a site asset, allowing you to use and reuse that one particular section of HTML and associated content passed through the template. The region definition defines which elements are available to use in a region template. It also defines what level of access the contributor has in switching the region content.

A region definition can contain more element definitions than are placed in the region template. Those element definitions listed in the region definition, but not used on the region template, still appear in Contributor when editing the data file. This is especially useful to remember if you use a "teaser" of information on one page to go to the full set of information on another page. This concept is covered in more detail in [Section 3.10, "Region Templates and Region Definitions."](#)

The boundaries of the contribution region are set by the placeholder. In a general sense, the placeholder is the area that is called the contribution region. The placeholder is not an asset - it is just a marker on a page template (or subtemplate) that is used to show where the contribution region is relative to the placement of the other objects on the site template or subtemplate. The placeholder definition assigned to a placeholder defines if the contributor has access to edit data, and if the contributor can switch the content.

The placeholder definition also defines which region definitions are available in the placeholder, and which region templates are available and which is the default for each region definition within the placeholder.

Since a placeholder does not have to be accessible to contributors, and since a placeholder can contain a subtemplate that is then divided into other placeholders, it's not always the case that a placeholder is strictly a contribution region. But in a general sense, when you conceive of a simple placeholder, it is the contribution region that the contributor works with.

8.4 Inserting Placeholders

Placeholders are used on page templates and subtemplates to mark an area to place the content. Placeholders are not assets; placeholder definitions are used to define how the placeholder is used on the page, and what content is placed on the page.

To insert the placeholder, follow these steps:

1. Click in the template where you want to place the placeholder.
2. On the toolbar, click the **Insert Placeholder** button.
3. The Define Placeholder dialog (see [Section A.75, "Define Placeholder Dialog"](#)) appears.

The dialog lists the placeholder names created in the Placeholder Definition Mappings dialog (see [Section A.73, "Placeholder Definition Mappings Dialog"](#)).

4. Select the placeholder name to associate with the placeholder and click **OK**.

8.5 Inserting Objects

Other site assets are easily added by placing the objects directly in the template. This can be done by selecting the icon from the toolbar, or by right-clicking on the template and adding the available site asset.

Adding an object to a page template, subtemplate, or region template is very simple.

To add an object, follow these steps:

1. Right-click in the template where you want to place the asset.
2. A contextual menu appears. Select **Insert Object**.
3. A further contextual menu opens, and those site assets that are available to insert are displayed.

For instance, you are not able to select an element unless you are in a region template.

4. After the site asset has been selected, it is added to the template.

When you add an asset that has dynamic content (such as a placeholder), then a marker appears in place of what content might be there. When you add an asset that has static content, such as an image, then the content appears.

An asset can also be added directly in source view.

On a region template, the region definition limits which elements are available to add.

8.6 Inserting Fragments

Fragments are simple to add to any template. To add a fragment to a template, you select the fragment from the fragment toolbox, and it is inserted at the selected location in the template.

However, there is one important consideration to use with a fragment.

A number of the fragments that are shipped with Oracle Site Studio Designer are composed of snippets that are used in the head and the body of the page template. Since subtemplates and region templates do not have a head, only a body, then the fragments do not work properly if inserted on these templates.

For you to have fragments that work properly in a subtemplate or a region template, you should create your own fragments that have no head snippet, and one body snippet. Fragments that have a head snippet and body snippet will work in a page template, but will not work in other templates. This limits the reusability of the fragment.

For more information, see [Chapter 13, "Working With Fragments."](#)

8.7 Working With Text

Working with text in Design view is much like working with text in a word processing program (with word-wrapping, line breaks, formatting options, and so on). To add text to a page template, simply click in the page template where you would like to add the text and start typing. Designer takes care of the necessary HTML code behind the scenes.

After you add and edit your text, you can format it by changing the typeface and font size, indenting or centering the text, adding bulleted or numbered lists, and so on.

Adding text directly to a template is discouraged. When the text is on the template, rather than the contributor data file, this makes localization more difficult.

8.8 Arranging Objects on the Template

How you arrange the information on your page, if you need a specific layout structure, can be done either with HTML tables or the use of CSS to place objects.

For the most part, when you insert an asset on a page template, subtemplate, or region template, it is inserted where you place the cursor. However, this does not necessarily mean that the asset appears in the finished web page the same way it appears in Designer. Variances such as the consumer's browser choice, the browser window size, and what other objects a contributor has placed in the contributor data file can make the web site appear to be organized differently.

Using CSS is the more efficient method, because CSS files are a managed object and can allow for maximizing reusability. If you place a table in an asset, such as a subtemplate, then to use that same structure in other assets, you must re-create the table. If you do use tables in this way, then you must make the changes across multiple places when you want to make the changes.

With CSS, you can identify the exact placement of each site asset, including elements, within the style sheet.

Many prefer to use tables because they are easy to set up and create on the web page. A table offers a quick solution of structuring the relative placement. With a table, you can create a shape of cells that place data into relative placement to each other.

8.9 Applying CSS Classes

Oracle Site Studio, by default, applies font and paragraph settings directly (inline) to the text.

Alternatively, you can apply classes from a cascading style sheet to your text (and images). Using CSS classes, you can store all of your font and paragraph settings in one location and then apply those settings to your web page.

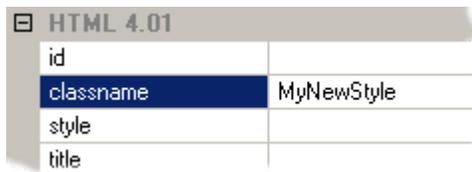
This saves you time, because rather than change the font and paragraph settings for each individual heading or paragraph, you can specify a class for each one. If you must change those settings, simply update the class.

To implement CSS classes, you insert a reference to your style sheet in the head of your page template. You can do this using a relative path, for example:

```
<link
href="<!--$HttpRelativeWebRoot-->groups/public/documents/adacct/stylessheet.css"
type="text/css" rel="stylesheet" />
```

Or, more typically, you put them in a fragment. Placing CSS references in a fragment does not expose the directory structure of your server.

Then, apply the class names from the style sheet to your text and graphics by editing the page template in Source view or by using the Properties pane in Design view.

Figure 8–6 Using the Properties Pane to Apply the Class "MyNewStyle."

The Site Assets pane in Designer can be very useful for adding, editing, and identifying the path to the Cascading Style Sheet in the content server.

Many of the fragments that come with Oracle Site Studio contain CSS files as an asset to the fragment (see [Section 13.12.5, "Adding, Editing, and Deleting Fragment Assets"](#)). You can reference class names from those CSS files, too, if you like.

You may also want to make these classes available to contributors while working in the Contributor application. This helps you enforce consistency across the site and make site-wide style updates much easier.

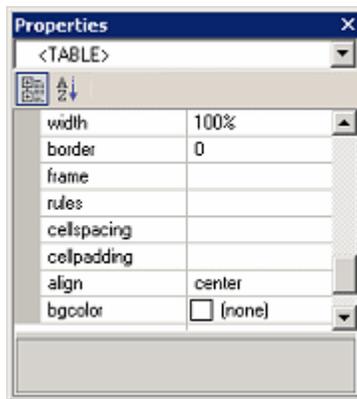
8.10 Working With HTML Tables

HTML tables are often used in web page design. They can be used to structure a web page when you do not want to use cascading style sheets to do so. The treatment of HTML tables in a page template is no different from tables in other web pages.

There are two principal uses for an HTML table: (1) to present data in a tabular format, and (2) to lay out text and graphics on a web page. As the designer of the site, you may find yourself using tables more frequently for the latter.

Working in a table, you can control how content appears on the web page by specifying the size and location of the table, the size of each column and row, and the content that goes into each cell. You keep your site visitors from seeing the table by hiding its borders. In Oracle Site Studio, tables are especially useful for positioning contribution regions and the elements within them.

To make specific edits to your table, you can modify the HTML `<TABLE>` tag directly in the Properties pane. You can also make changes in source view.

Figure 8–7 Properties of the `<TABLE>` Tag in Design View

In the Properties pane, you can change the background color, border width, cell padding, and several other settings. All of the attributes that would be entered into the `<TABLE>` tag are available in the Property Panel. For a complete description of the

<TABLE> tag, visit the World Wide Web Consortium (W3C) web site:
<http://www.w3.org/TR/html401/struct/tables.html#edef-TABLE>.

8.11 Viewing Templates in Designer

There are three ways to view a template while in Designer: Design view, Source view, and the preview. Each offers a different way to look at each template while working with that template in Designer.

In each template window in the Designer workspace, there will be three tabs at the bottom. Select the tab to choose with the Design view, the Source view, or the Preview for that template.

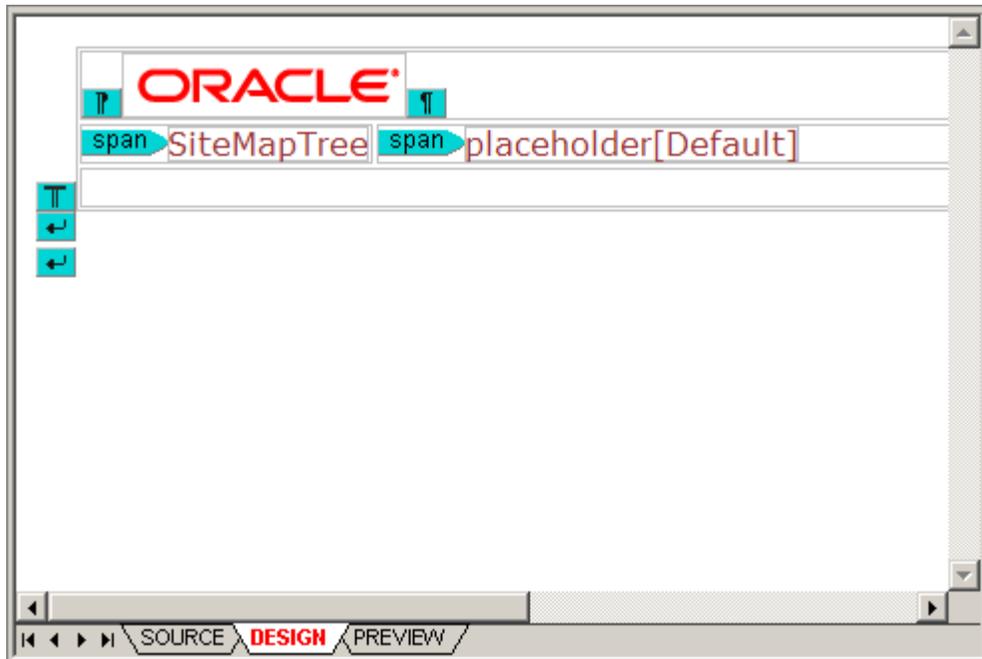
This section covers these topics:

- [Section 8.11.1, "Design View"](#)
- [Section 8.11.2, "Source View"](#)
- [Section 8.11.3, "Preview"](#)

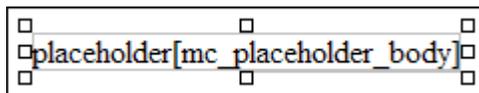
8.11.1 Design View

When you open a template (page template, region template, or subtemplate) in Designer, it opens in the workspace in design view, where you can edit the template as needed. The workspace then functions as an editor that offers many common editing features.

Design view displays all objects as they are arranged on the template. In design view you will see the positioning of both static content (for example, a fixed banner graphic) and dynamic content (for example, placeholders and fragments). Depending on how the template was set up and what positioning methods are used (for instance, using tables or CSS to align assets), this view may approximate the page layout when viewed in a web browser. You see text, images, colors, hyperlinks, and tables as they appear on the actual web site, but dynamic content (from fragments and scripts) is represented as tags rather than the actual content. This is because most fragments and scripts require server-side scripting and dynamically rendered content.

Figure 8–8 Design View

You can edit the text and objects using common editing keystrokes (for example, Ctrl+C and Ctrl+V to copy and paste), the application toolbars (see [Section 5.10, "Toolbars"](#)), and right-click menus (see [Section 5.6.7, "Right-Click Menus"](#)). To select an object, simply click it, so it displays selection handles ([Figure 8–9](#)).

Figure 8–9 Selected Object in Design View

For edits that cannot be performed using design view, you can use the properties pane (see [Section 5.5, "Properties Pane"](#)). If you cannot make a specific edit here, you can always edit the actual code in source view.

Designer can also highlight the location of HTML tags in design view. To turn this feature on, click the **Show/Hide HTML Tags** icon ([Figure 8–10](#)) in the formatting toolbar. You can click the down arrow next to the icon to show or hide specific HTML tags.

Figure 8–10 Show/Hide HTML Tags

8.11.2 Source View

Source view displays the code that is in the template: HTML, XML, JavaScript, Idoc Script, Oracle Site Studio tags, and so on.

Figure 8–11 Source View

```

<body>
  <p>
    <tbody>
      <tr>
        <td width="20%">
          <!--SS_BEGIN_SNIPPET(fragment1,2)--><!--$ssFragmentInsta
        </td>

        <td><!--$wcmPlaceholder("Default")--></td>
      </tr>
    </tbody>
  </table><br />
  <br />
</body>
</html>

```

The information in source view appears as color-coded text:

- Black is used for XML code, HTML tag attributes, and text that appears on the web page.
- Purple is used for HTML tags.
- Blue is used for HTML tag attribute values.
- Green is used for Idoc Script, HTML comments, and code inserted by Oracle Site Studio.

You can change many of these settings, including the typeface, indentation levels, and line wrapping. For more information, see [Section 6.9, "Formatting the Code in Source View."](#)

Source view offers complete control over the site template. This can be especially true if you are implementing several custom assets. In fact, you may find yourself starting off in design view to create the page and then switching to source view to customize the appearance and behavior of the page.

8.11.3 Preview

Preview provides an actual view of a web page with all assets in place, as it appears in a web browser for consumers. This is useful for previewing templates and checking how fragments and placeholders are positioned on the template (which may be hard to see in design mode, especially if the page layout is controlled by CSS).

Figure 8–12 Preview



Note: Oracle Site Studio requires Internet Explorer 7 or higher to provide previewing capabilities.

8.12 Region Templates and Region Definitions

Region definitions specify what contribution regions on a web page contain, whereas region templates define what contribution regions look like. In other words, region definitions specify the structure (and attributes) of site content, and region templates define the visual presentation of that content on a web page.

There can be multiple region templates for each region definition. This allows site content to be displayed differently depending on the context within the site. If there are multiple region templates for a region definition, then the default region template is used unless a different one was specifically set to be used.

Since the creation of a region template requires you associate it with a region definition, it is recommended that you create region definitions first, then create region templates.

8.13 Working With Native Documents and Conversion Definitions

Native documents require that a conversion definition is available to define the rules and templates in place to convert the native document on the web page. To use and convert native documents, you must first have Check Out and Open and Dynamic Converter installed on the content server.

The dynamic conversion can only be located on a region template. Adding the conversion to the region template is simple, and works the same way that adding any other asset to the template would.

To add the dynamic conversion to the region template, follow these steps:

1. Open the region template that you want the converted document to appear on.

2. Right-click and from the menu select **Insert Object**, and then select **Dynamic Conversion**.

The Insert Dynamic Conversion dialog (see [Section A.78, "Insert Dynamic Conversion Dialog"](#)) opens.

3. Select the dynamic conversion to use.
4. Click **OK**.

For more information on native documents and conversion definitions, see [Section 10.8, "Working With Native Documents."](#)

Working With Definitions

Oracle Site Studio definitions - placeholder definitions, region definitions, and element definitions - control which data is presented, and additionally control how the contributor interacts with the data.

This section covers the following topics:

- [Section 9.1, "About Definitions"](#)
- [Section 9.2, "Creating a New Definition"](#)
- [Section 9.3, "Editing Definitions"](#)
- [Section 9.4, "Copying a Definition"](#)
- [Section 9.5, "Viewing the Content Information of a Definition"](#)
- [Section 9.6, "Adding a Definition to a Site"](#)
- [Section 9.7, "Removing a Definition from a Site"](#)
- [Section 9.8, "Working With Placeholder Definitions"](#)
- [Section 9.9, "Working With Region Definitions"](#)
- [Section 9.10, "Working With Element Definitions"](#)
- [Section 9.11, "Viewing Definitions in Designer"](#)
- [Section 9.12, "Definition Bundles"](#)

9.1 About Definitions

When a definition is used, it then prescribes what can be contained in the asset the definition is associated with. Placeholder definitions specify which region templates, region definitions, and subtemplates can appear in a placeholder, region definitions specify which elements can appear in a region template, and element definitions control how the contributor can modify the data within the contributor data file.

This section covers the following topics:

- [Section 9.1.1, "About Placeholder Definitions"](#)
- [Section 9.1.2, "About Region Definitions"](#)
- [Section 9.1.3, "About Element Definitions"](#)

9.1.1 About Placeholder Definitions

Placeholder definitions are what connect each placeholder to the content and other assets associated with that place holder to display. Since the placeholder is simply a

marker on the page, the placeholder definition is used to determine whether a contributor can edit the content displayed within the placeholder, if they can view reports, and other actions.

Additionally, the placeholder definition includes the region definitions that can be used within the area of the placeholder (also called a contribution area), and these region definitions are then connected to one or more region templates that are also controlled in their availability by listing them in the placeholder definition.

A subtemplate, which is a section of HTML that may include multiple placeholders (and by association other placeholder definitions), can also be listed as available in the placeholder definition. A subtemplate can be used in the contribution region instead of a region definition and a region template.

The placeholder definition controls which region definitions are available within the placeholder. By including more region definitions in the placeholder definition, then you have more options to use the region definition as a method of controlling the data file assigned to the contribution region.

Templates, such as region templates, subtemplates, and page templates, are used to control how the associated data is displayed. Definitions, such as region definitions, placeholder definitions, and element definitions, control what data is displayed, and how the contributor interacts with that data.

For more information, see [Section 9.8, "Working With Placeholder Definitions."](#)

9.1.2 About Region Definitions

The typical region definition contains multiple elements, and various combinations of these elements are used on several different region templates. Each region definition contains a list of the elements available to place on the associated region template. This enables multiple region templates to access the same grouping of elements (but it does not have to be the same data file) and arrange them in different ways.

Because the region definitions limit the availability of elements in the contribution model, it is especially important to follow good naming conventions for region definitions. It is possible that you have more region definitions than any other site asset. A good naming convention helps you sort out which definitions are best suited for the different region templates you create without having to open multiple region definitions and see what exactly they do.

Since the creation of a region template requires you associate it with a region definition, it is recommended that you create region definitions first, then create region templates.

For more information, see [Section A.42, "Region Definition Dialog,"](#) and [Section 9.9, "Working With Region Definitions."](#)

9.1.3 About Element Definitions

Elements are the objects used to define how the contributors can edit the data. The available elements are WYSIWYG (What You See Is What You Get), plain text, image only, static lists, and dynamic lists. You can also create custom elements that can be used for several situations, including processing data-driven site creation.

9.2 Creating a New Definition

To create a definition, perform these tasks:

1. In the menu on the Site Assets pane, select the definition type.
2. Click the **Create New** icon (Figure 9–1), select **New**, and then the definition type. The Assign Info Form opens for you to check the item into the content server.

Figure 9–1 Create New Icon



3. Enter appropriate values for the Assign Info Form. For guidelines on naming Oracle Site Studio assets, see [Section 4.2, "Naming Site Assets."](#)
4. When complete, click **Assign Info**.
5. The definition is created.

9.3 Editing Definitions

Editing each definition uses different interfaces. Each interface maximizes the ease in making the changes specific to the type of definition you are working with.

This section includes the following topics:

- [Section 9.3.1, "Editing a Placeholder Definition"](#)
- [Section 9.3.2, "Editing a Region Definition"](#)
- [Section 9.3.3, "Editing an Element Definition"](#)

9.3.1 Editing a Placeholder Definition

The placeholder definition should contain the region definitions, region templates, and subtemplates that would be used to control the content used in the contribution region.

To edit a placeholder definition, perform these tasks:

1. In the menu on the Site Assets pane, select **Placeholder Definitions**.
2. From the list, select the placeholder definition to be edited and click the **Edit** icon (Figure 9–2).

The placeholder definition dialog opens. For more information, see [Section A.72, "Placeholder Definition Dialog,"](#) and [Section 9.8, "Working With Placeholder Definitions."](#)

Figure 9–2 Edit Icon



3. Enter a description for the placeholder definition. You should use a good naming convention for placeholder definitions. If you can understand what the placeholder definition does from the name, then you won't have to view the definition each time you consider using it.
4. Select or deselect the actions you want to allow with the placeholder definition. All of these actions impact how the contributor works with the contribution region.

5. In the allowed region definitions, click **Add** to add a region definition, and a region template.
6. In the allowed subtemplates, click **Add** to add a subtemplate.
7. Click **Remove** in any section to remove the selected region definition, region template, or subtemplate, as applicable.

When editing the placeholder definition, you should keep the region definitions, region templates, and subtemplates in mind.

For more information, see [Section A.72, "Placeholder Definition Dialog,"](#) and [Section 9.8, "Working With Placeholder Definitions."](#)

9.3.2 Editing a Region Definition

When the region definition has been created, you can open it to include the element definitions you need.

To edit a region definition, perform these tasks:

1. In the menu on the Site Assets pane, select **Region Definition**.
2. From the list, select the region definition to be edited and click the **Edit** icon ([Figure 9-3](#)).

The region definition dialog opens. For more information, see [Section A.42, "Region Definition Dialog."](#)

Figure 9-3 Edit Icon



3. Enter a description for the region definition.
4. Click **Add** to add elements to the region definition.

The element dialog opens.

The Name you give the element is the name used to identify the element within the region definition, and in any other site assets that refer to this region definition.

The Label you give the element definition is the name of the element (located in the toolbar) in Contributor.

The Info you enter for the element you add opens as a tooltip when the contributor hovers the mouse over the Label.

5. Select an element and click **Remove** to remove an element.
6. Select an element and click Move Up or Move Down to change the order of elements in the region definition.

The order of elements is the order that they display in when Contributor is opened for the contribution region based on this region definition.

In this dialog, you should include all of the elements you want to include in the region definition. You can use and place these element definitions in the region template when the region template and region definition are associated.

For more information, see [Section 9.9, "Working With Region Definitions."](#)

9.3.3 Editing an Element Definition

To edit an element definition, perform these tasks:

1. In the menu on the Site Assets pane, select **Element Definitions**.
2. From the list, select the element definition to be edited and click the **Edit** icon (Figure 9-4).

A dialog opens that enables you to edit the selected element definition. Depending on the element type, one of these screens opens:

- [Section A.27, "WYSIWYG Element Definition Dialog"](#)
- [Section A.29, "Plain Text Element Definition Dialog"](#)
- [Section A.31, "Image Element Definition Dialog"](#)
- [Section A.33, "Custom Element Definition Dialog"](#)
- [Section A.36, "Static List Element Definition Dialog"](#)
- [Section A.40, "Dynamic List Element Definition Dialog"](#)

Figure 9-4 Edit Icon



3. Enter the info for the element definition. This is a short description of the element to help you remember how the element definition was set up. The information you provide here helps you remember, at a glance, which specific settings you have placed in the element definition. This is especially helpful if you have created multiple similar element definitions with only slight differences in the actions from each other.
4. Select or deselect the actions you would like to allow the contributor to have for the selected element definition.

If you are creating a custom element, the process is different.

In each case, the main focus of the editing is to select which objects are available to the contributors on the toolbars. Each element you define has its own toolbar, and what appears on the toolbar (for example, toolbar buttons allowing changes to font size, font style, paragraph formatting, and so forth) is defined when you select or clear the checkboxes in the actions list of an element definition.

It is recommended that you have multiple definitions of each type, to allow for different toolbars that the users can have in different instances. For specific information on the actions available to modify in each element definition, see the specific screen listed above.

9.4 Copying a Definition

It is simple to copy a definition from one you have already made, allowing you to modify one or two aspects of it to make asset creation go much faster.

To copy a definition, follow these steps:

Copying a Definition from the list

1. In the menu on the Site Assets pane, select the definition type.
2. Select the definition you want to copy from the list.

3. Click the **Create New** icon (Figure 9–5), select **Copy**, and then **Selected**:
The Assign Info Form opens for you to check the item into the content server.

Figure 9–5 Create New Icon



4. Enter appropriate values for the Assign Info Form.
5. When complete, click **Assign Info**.
6. The definition is copied.

When the definition has been copied, it can then be opened to edit.

Copying a Definition from the server

1. In the menu on the Site Assets pane, select the definition type.
2. Click the **Create New** icon (Figure 9–5), select **Copy**, and then **from Server**.
A search results page opens.
3. Click the corresponding **Select** button of the ones you want to copy.
The Assign Info Form opens for you to check the item into the content server.
4. Enter appropriate values for the Assign Info Form.
5. When complete, click **Assign Info**.
6. The definition is copied.

Copying a Definition from your local instance

1. In the menu on the Site Assets pane, select the definition type.
2. Click the **Create New** icon (Figure 9–5), select **Copy**, and then **from Local**.
An open file dialog opens.
3. On your local instance, navigate to the definition you want to copy.
4. Select the definition and click **Open**.
The Assign Info Form opens for you to check the item into the content server.
5. Enter appropriate values for the Assign Info Form.
6. When complete, click **Assign Info**.
7. The definition is copied.

9.5 Viewing the Content Information of a Definition

To view content information for a definition, perform these tasks:

1. In the menu on the Site Assets pane, select the definition type.
2. Select a definition from the list.
3. Click the **Doc Info** icon (Figure 9–6).

The content information page opens.

Figure 9–6 Doc Info icon

9.6 Adding a Definition to a Site

To add a definition to a site, perform these tasks:

1. In the menu on the Site Assets pane, select the definition type.
2. Click the **Add to Site** icon (Figure 9–7).

A search results page opens.

Figure 9–7 Add to Site Icon

3. Select the definition or definitions to add, click **Oracle Site Studio**, and then **Select Marked Documents**.
4. Depending on your configuration, you may receive a caution that you are about to add existing asset(s) to your site. Click **OK**.

The definition is now associated with the web site, and you can now open the definition to edit.

9.7 Removing a Definition from a Site

To remove a definition from a site, perform these tasks:

1. In the menu on the Site Assets pane, select the definition type.
2. From the list, select the definition to be removed and click the **Remove From Site** icon (Figure 9–8).

When you remove a definition, you are simply removing it from the site, not deleting from the content server.

Figure 9–8 Remove from Site Icon

9.8 Working With Placeholder Definitions

When you insert a placeholder on a page template, it is simply a named marker where content displays. The placeholder definition controls how the data opens in the placeholder. Placeholder definitions also control if the contributor can approve or reject the assets in workflow, if the contributor can view web reports, if the contributor can edit the assets in the placeholder, and more.

The relationship between the placeholder name and the placeholder definition is called the mapping. These mappings are used to determine the placeholder definition used as the default when you create a new placeholder.

The mappings can be done in four ways. A default mapping is defined in the web site properties under the Default Placeholder Definition. If a mapping is made here, then

it is the one used between a placeholder definition and a placeholder name if none of the other three mappings have been defined.

A global mapping, made in the Placeholder Definition Mappings dialog (see [Section A.73, "Placeholder Definition Mappings Dialog"](#)), establishes a default mapping for the placeholder name throughout the site. The placeholder name can be used on multiple templates in multiple sections, and the same mapping applies.

A section property mapping is defined in the section properties under the Primary Page Placeholder Definition Mappings and Secondary Page Placeholder Definition Mappings. The section this is done in now uses a different mapping from any other section where the placeholder is used. The section property mapping overrides a global mapping and a default mapping, but not a mapping in a tag.

The fourth method of creating a mapping is by defining the mapping in the placeholder tag itself. This type of mapping overrides all other mappings, including global mappings. When you define a mapping in the placeholder tag, you are making a change to the placeholder tag in source view. When you use a mapping this way, this means that every time the page template is used, the placeholder on the template always uses the specified placeholder definition regardless of where in the web site the page template is used.

What the contribution region identified by the placeholder contains, and how it looks on the site, is defined using region templates and region definitions (see [Section 3.10, "Region Templates and Region Definitions"](#)). A page template may contain multiple placeholders, each of which represents a contribution region on the page. There are no files associated with placeholders; that is, there are no "placeholder files" on the content server.

This section covers the following topics:

- [Section 9.8.1, "Specifying a Default Placeholder Definition"](#)
- [Section 9.8.2, "Creating a Global Placeholder Definition Mapping"](#)
- [Section 9.8.3, "Creating a Section Property Placeholder Definition Mapping"](#)
- [Section 9.8.4, "Specifying a Placeholder Definition Directly in the Placeholder Tag"](#)
- [Section 9.8.5, "Associating a Placeholder with a Global Placeholder Definition"](#)
- [Section 9.8.6, "Adding a Region Definition to a Placeholder Definition"](#)
- [Section 9.8.7, "Associating a Region Template with a Placeholder Definition"](#)
- [Section 9.8.8, "Associating a Subtemplate with a Placeholder Definition"](#)

9.8.1 Specifying a Default Placeholder Definition

The default placeholder definition is the placeholder definition used when no other mappings apply. It is considered more of a "catch-all."

To specify a default placeholder definition, follow these steps:

1. In the Web Site Properties, click in the Default Placeholder Definition.
2. Click the additional information icon that appears.
A content server search opens, showing the available placeholder definitions.
3. Select the placeholder definition to use as the default.
4. The default placeholder definition appears in the web site properties.

9.8.2 Creating a Global Placeholder Definition Mapping

The list of available global placeholder mappings are defined on the Placeholder Definition Mappings dialog (see [Section A.73, "Placeholder Definition Mappings Dialog"](#)), which is available by selecting **Tools** from the menu bar, then selecting **Define Placeholder Definition Mappings**. The placeholder mappings allow a placeholder to have a different definition depending on whether the placeholder is used on a primary or a secondary page. The mappings are stored in the project file for the web site.

The global placeholder definition mapping overrides the mapping created as the default placeholder definition in the web site properties (see [Section 9.8.1, "Specifying a Default Placeholder Definition"](#)).

To create a global placeholder mapping, follow these steps:

1. From the menu bar, select **Tools**, and then **Define Placeholder Definition Mappings**.

The Placeholder Mappings dialog opens (see [Section A.73, "Placeholder Definition Mappings Dialog"](#)).

2. Click **Add** to create a new mapping.

The Define Placeholder Definition Mapping dialog opens (see [Section A.74, "Define Placeholder Definition Mapping Dialog"](#)).

3. Enter a placeholder name that is easy to recognize, and also helps you understand what the mapping does. In the two menus, enter a placeholder definition that the placeholder uses on a primary page and on a secondary page. You can enter for either a primary page or a secondary page, or both.
4. Click **OK**.
5. Your placeholder definitions now display in the Placeholder Mappings dialog, and are available to select when you insert a placeholder.

9.8.3 Creating a Section Property Placeholder Definition Mapping

The section property placeholder definition mapping lists the mapping between a placeholder name and a placeholder definition. The section this mapping is in uses a different mapping from any other section where the placeholder is used.

The section property placeholder definition mapping overrides the global placeholder definition mapping and the default placeholder definition.

There are two ways to make this section property, by using a dialog, and by entering the placeholder name and placeholder definition directly in the section properties. The dialog uses a friendlier user interface.

To create a section property placeholder definition mapping using the dialog, follow these steps:

1. Right-click a page template within the appropriate section in the hierarchy. In the menu, select **Placeholder Section Properties...**

The Placeholder Section Properties dialog opens (see [Section A.70, "Placeholder Section Properties Dialog"](#)).

2. Click **Add**.

The Edit Placeholder Section Properties dialog opens (see [Section A.71, "Edit Placeholder Section Properties Dialog"](#)).

3. Enter the Name, and the placeholder definition ID . If you want to include a region template, subtemplate, or contributor data file in the section property, add those as well.
4. Click **OK** to close the Edit Placeholder Section Properties dialog.
The section property placeholder name mapping is created.
5. Click **OK** to close the Placeholder Section Properties dialog.

To create a section property placeholder definition mapping directly in the section properties, follow these steps:

1. View the section properties for the section you want to create the mapping in.
2. In the Primary Page Placeholder Definition Mappings, enter the placeholder name and placeholder definition in this format:

```
name=contentID
```

where the name is the placeholder name, and the content ID is the content ID of the placeholder definition. To enter multiple placeholder names and placeholder definitions, use an ampersand ('&') between each pairing, in this manner:

```
name1=contentID1&name2=contentID2&name3=contentID3
```

3. For the secondary pages, enter a placeholder name and placeholder definition for the Secondary Page Placeholder Definition Mappings. Multiple name and definition pairings are allowed just as in the Primary Page Placeholder Definition Mappings.
4. The Primary and Secondary Page Placeholder Definition Mappings display in the section properties.

9.8.4 Specifying a Placeholder Definition Directly in the Placeholder Tag

The placeholder tag can be written so that the placeholder definition is specified explicitly in the placeholder tag. This direct specification is used each time the page template is used.

This method of correlating a placeholder definition and a placeholder tag overrides all other placeholder definition mappings.

To specify the placeholder definition directly in the placeholder tag, follow these steps:

1. Open the page template that contains the placeholder.
2. Click the **Source View** tab.
3. In the placeholder tag, enter the mapping in this manner:

```
<!--$wcmPlaceholder("Sales", "placeholderDefinitionDocName=placedef_
salescontact")-->
```

In this example, the placeholder name is Sales, and the placeholder definition content ID is placedef_salescontact.

4. The mapping is now created in the tag.

You can also assign a placeholder definition to be written in the tag from the Define Placeholder dialog (see [Section A.75, "Define Placeholder Dialog"](#)):

1. Open the page template that contains the placeholder.
2. Double-click the placeholder.

The Define Placeholder dialog opens.

You can also do this by inserting a new placeholder.

3. The name in the menu is the name of the placeholder you selected (unless the placeholder is new). Check the **Optionally specify the placeholder definition on the template** checkbox.

The dialog expands. Select the placeholder definition to map in the placeholder tag from the menu.

4. The mapping is now created in the tag.

9.8.5 Associating a Placeholder with a Global Placeholder Definition

When you have created the placeholder mappings, the placeholder definition is associated to placeholders you create according to the mappings.

By using a different placeholder definition in a placeholder, you can use different subtemplates, region templates, or allow for some instances of a placeholder to be available to contributors for editing, or any other aspect that has been defined in your placeholder definition.

To associate a placeholder with a global placeholder definition, follow these steps:

1. Insert the placeholder on the page template (see [Section 8.4, "Inserting Placeholders"](#))
2. As you insert the placeholder, the Define Placeholder dialog (see [Section A.75, "Define Placeholder Dialog"](#)) opens. Select a placeholder name.
3. Click **OK**.
4. The placeholder definition is now associated with the placeholder.

9.8.6 Adding a Region Definition to a Placeholder Definition

The placeholder definitions list the region definitions, region templates, and subtemplates associated (and thus available to use) with the placeholder. Adding a region definition to, or removing a region definition from, a placeholder definition is a very simple task.

When a region definition is associated with a placeholder, then only those elements that are associated with that region definition are available to use in Contributor. This is especially important to remember in those cases where you might give a contributor access to switch the region content or the region template.

To associate a region definition with a placeholder definition, follow these steps:

1. Open the placeholder definition from the Site Assets pane.
The Placeholder Definition dialog (see [Section A.72, "Placeholder Definition Dialog"](#)) appears.
2. In the Allowed Region Definitions pane, click **Add**.
The Select a Region Definition Dialog appears.
3. Select a region definition, and click **OK**.
4. The region definition is now associated with the placeholder definition.

When you have added a region definition, you may want to add a region template to associate with the definition.

To remove a region definition from a placeholder definition, follow these steps:

1. Open the placeholder definition from the Site Assets pane.
The Placeholder Definition dialog (see [Section A.72, "Placeholder Definition Dialog"](#)) appears.
2. In the Allowed Region Definitions pane, select a region definition.
3. Click **Remove**.
4. The region definition is now removed from the placeholder definition.

9.8.7 Associating a Region Template with a Placeholder Definition

The placeholder definition lists the region definitions, region templates, and subtemplates associated (and thus available to use) with the placeholder. Adding a region definition to, or removing a region definition from, a placeholder definition is a very simple task.

When a region template is associated with a placeholder, then the region template along with the region definition used, can define the element definitions which displays the data and be available in Contributor. Region templates are used to define a specific layout of elements; when a region template is added, it must be associated with a region definition already in the placeholder definition.

To associate a region template with a placeholder definition, follow these steps:

1. Open the placeholder definition from the Site Assets pane.
The Placeholder Definition dialog (see [Section A.72, "Placeholder Definition Dialog"](#)) opens.
2. In the Allowed Region Definitions pane, select a region definition that the region template is used with.
3. In the Allowed Region Templates pane, click **Add**.
The Select a Region Template Dialog opens.
4. Select a region definition, and click **OK**.
5. The region definition is now associated with the placeholder definition.

Only those region templates that are associated to a region definition display in the Select Region Template dialog. The association between a region template and a region definition is stored as metadata on the region template.

To make a region template the default, complete these steps:

1. Open the placeholder definition from the Site Assets pane.
The Placeholder Definition dialog (see [Section A.72, "Placeholder Definition Dialog"](#)) opens.
2. In the Allowed Region Definitions pane, select a region definition.
The Allowed Region Templates pane displays the associated region templates.
3. In the Allowed Region Templates pane, select a region definition.
4. Click **Make Default**.
5. The region template is now the default template for the selected region definition within this placeholder definition.

To remove a region template from a placeholder definition, follow these steps:

1. Open the placeholder definition from the Site Assets pane.
The Placeholder Definition dialog (see [Section A.72, "Placeholder Definition Dialog"](#)) appears.
2. In the Allowed Region Templates pane, select a region template.
3. Click **Remove**.
4. The region template is now removed from the placeholder definition.

9.8.8 Associating a Subtemplate with a Placeholder Definition

The placeholder definition lists the region definitions, region templates, and subtemplates associated (and thus available to use) with the placeholder. Adding a subtemplate to, or removing a subtemplate from, a placeholder definition is a very simple task.

When you use a subtemplate associated with a placeholder, the site assets, and any placeholders on the subtemplate, appear on the final web page. It is not necessary to use a subtemplate.

A subtemplate can be any section of HTML comprised of static assets and placeholders, each with their own definitions. This is especially important to remember in those cases where you might give a contributor access to switch the region content or the region template.

To associate a subtemplate with a placeholder definition, follow these steps:

1. Open the placeholder definition from the Site Assets pane.
The Placeholder Definition dialog (see [Section A.72, "Placeholder Definition Dialog"](#)) appears.
2. In the Allowed Subtemplates pane, click **Add**.
The Select a Subtemplate dialog appears.
3. Select a subtemplate, and click **OK**.
4. The subtemplate is now associated with the placeholder.

To remove a subtemplate from a placeholder definition, follow these steps:

1. Open the placeholder definition from the Site Assets pane.
The Placeholder Definition dialog (see [Section A.72, "Placeholder Definition Dialog"](#)) appears.
2. In the Allowed Subtemplates pane, select a subtemplate.
3. Click **Remove**.
4. The subtemplate is now removed from the placeholder definition.

9.9 Working With Region Definitions

Each region definition contains a list of associated element definitions. This list of definitions is the complete list of elements available to use in the layout of the region template, after the region template and region definition are associated.

A region template is not required to use all elements available in a region definition. Usually, not all elements are used to display in the region template. However, after Contributor is launched, all elements display in the editor, whether they are displayed in the associated region template or not.

The association between a region template and a region definition is stored in the xRegionDefinition metadata for the region template.

Associating an element definition with a region definition

The region definitions list the element definitions available to use with the region template. Adding an element definition to, or removing an element definition from, a region definition is a very simple task.

To associate an element definition with a region definition, follow these steps:

1. Open the region definition from the Site Assets pane.
The Region Definition dialog (see [Section A.42, "Region Definition Dialog"](#)) appears.
2. Click **Add**.
The Element Dialog (see [Section A.43, "Element Dialog"](#)) appears.
3. Enter the information for the element (using a good naming convention to easily recognize the element definition as it is used), select the element definition from the menu, and click **OK**.
4. The element definition is now associated with the region definition.

It is also possible to embed the element definition in the region definition. When you embed the element definition, you are creating a singular instance of the named element definition within the region definition, which is not affected by global changes in the way that including the element definition by reference would.

To embed an element definition in a region definition, add it as you would normally, but on the Element Dialog (see [Section A.43, "Element Dialog"](#)), check the **Embed the element definition inside the region definition** checkbox.

To remove an element definition from a region definition, follow these steps:

1. Open the region definition from the Site Assets pane.
The Region Definition dialog (see [Section A.42, "Region Definition Dialog"](#)) appears.
2. Select an element definition. Click **Remove**.
3. The element definition is now removed from the region definition.

Region definitions define the types of content used on a web site. They are essentially groups of individual elements which define the various chunks of reusable information for a particular site content type.

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.

9.10 Working With Element Definitions

Each element definition is used to specify a toolbar, and any customization scripts and validation scripts you may want to include. These scripts are included when creating each element definition.

Elements are controlled by **element definitions**. An element is basically an instantiation of its associated element definition, which specifies the editing experience available to contributors for the element on the web page (specifically, what editing features are available in the Contributor editor). Element definitions are individually managed site assets, which means that they can be reused within a web site, or even across web sites (providing all sites are managed on the same content server). Elements of the same type may have different element definitions associated with them to provide different editing environments for contributors depending on the context in which the element is used. Similarly, multiple elements may share the same element definition, providing the same editing experience to contributors for each of the elements.

9.11 Viewing Definitions in Designer

There are two ways to view a definition while in Designer: Source view, and Form view. Each offers a different way to look at each definition while working with that template in Designer.

In each definition window in the Designer workspace, there will be two tabs at the bottom. Select the tab to choose the Source view or the Form view for that definition.

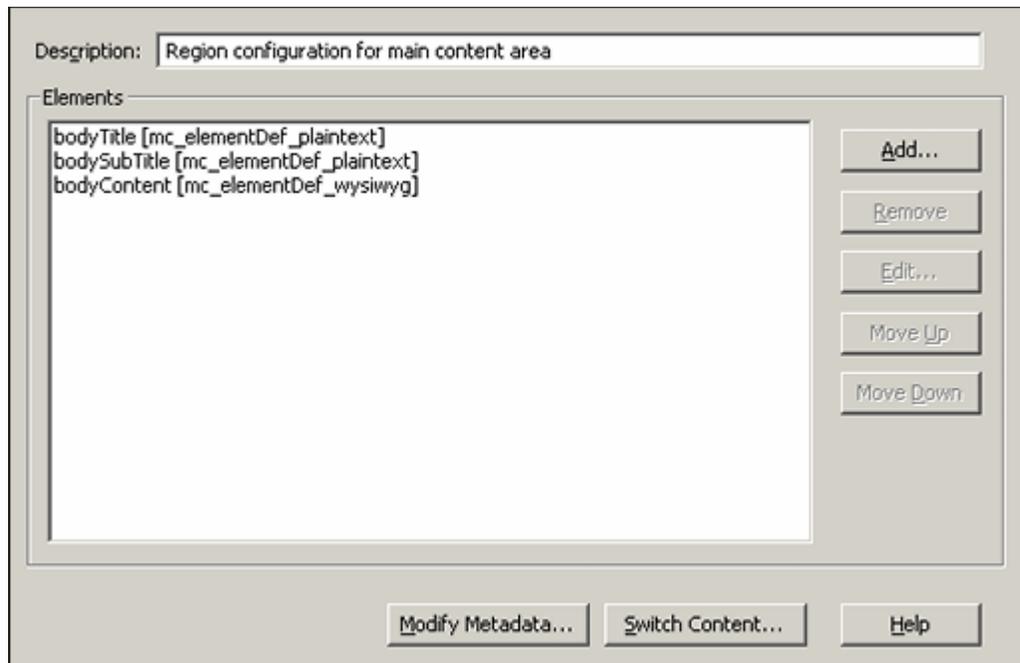
This section covers these topics:

- [Section 9.11.1, "Form View"](#)
- [Section 9.11.2, "Source View"](#)

9.11.1 Form View

Form view is a graphical user interface for working with a definition. Using the form view, the different options for each definition are easily modified within the dialog. All options available to modify in form view are also available in source view.

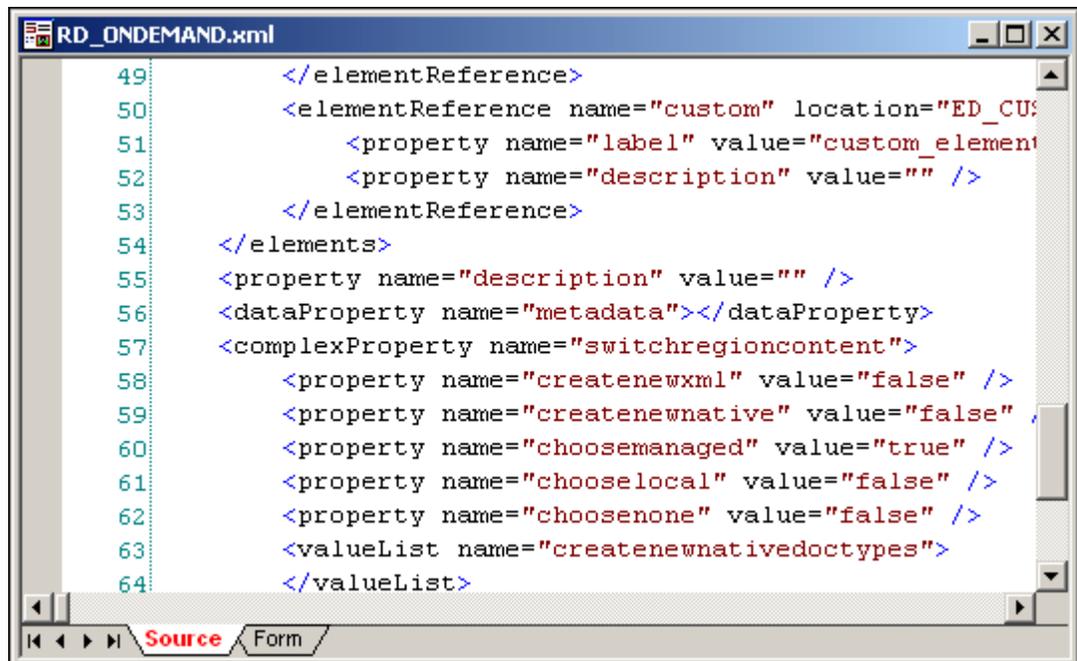
Form view provides a dialog in which you can set the properties of definitions; for example, element definitions, region definitions, placeholder definitions, and conversion definitions. [Figure 9-9](#) shows an example of a region definition in form view. Most form views have a **Help** button that you can click to see context-sensitive help information for the current form.

Figure 9–9 Form View (Region Definition)

Screenshot of a region definition in form view, which provides an interface where you can manage the region definitions and set their properties.

9.11.2 Source View

Source view displays the code for the definition.

Figure 9–10 Source View

The information in source view appears as color-coded text:

- Black is used for XML code, HTML tag attributes, and text that appears on the web page.
- Purple is used for HTML tags.
- Blue is used for HTML tag attribute values.
- Green is used for Idoc Script, HTML comments, and code inserted by Oracle Site Studio.

You can change many of these settings, including the typeface, indentation levels, and line wrapping. For more information, see [Section 6.9, "Formatting the Code in Source View."](#)

Source view offers complete control over the site template. This can be especially true if you are implementing several custom assets. In fact, you may find yourself starting off in design view to create the page and then switching to source view to customize the appearance and behavior of the page.

9.12 Definition Bundles

Definition bundles are bundles of site assets; typically a definition and related templates. However, it can be a varying combination of definitions and templates depending on how the bundle is constructed.

This section covers the following topics:

- [Section 9.12.1, "About Definition Bundles"](#)
- [Section 9.12.2, "Downloading Bundles"](#)
- [Section 9.12.3, "Uploading Bundles"](#)

9.12.1 About Definition Bundles

Definition bundles are zipped (compressed) collections of definitions (and related templates, if applicable). A definition bundle is a simple method to take a related grouping of definitions and copy them to another web site, or to a completely different content server.

The definition bundle includes the selected definition, and the related definitions below the selected definition in the hierarchy. If you bundle a selected region definition, then it will include all associated region templates, and element definitions. If you select a placeholder definition, it will bundle all subtemplates, region definitions, region templates, and element definitions associated with the placeholder definition.

9.12.2 Downloading Bundles

When you choose to download a bundle, it is automatically zipped for you, and you will download the zipped file. There are two ways you can download a definition bundle.

You can download a definition bundle based on a selected definition directly from the site assets pane, or you can use the Oracle Content Server to select a definition and download the bundle based on that definition.

Downloading from the Site Assets Pane

To download a definition bundle from the site assets pane, perform these steps:

1. Select a definition (either placeholder definition, region definition, or element definition) in the site assets pane.
2. Right-click the definition and choose **Download Definition Bundle..**
The save dialog appears.
3. Name and save the bundle to the local instance.

Downloading from the Menu Bar

To download a definition bundle from the menu bar, perform these steps:

1. From the menu bar, choose **File**, then **Definitions**, then **Download**.
2. Select either **Placeholder Definition Bundle**, **Region Definition Bundle**, or **Element Definition Bundle**.
Oracle Content Server search results are displayed based on the selected definition type.
3. Select the definition to create and download the bundle from.
The save dialog appears.
4. Name and save the bundle to the local instance.

9.12.3 Uploading Bundles

When you upload a bundle in Designer, it is automatically unpacked and associated with the web site you had open in Designer when you uploaded.

The bundle, when packed and downloaded, has the web site you bundled it from still associated with it in the *xWebsites* metadata field (for more information on metadata, see *Technical Reference for Oracle Site Studio*). The site that is open in Designer will have its metadata value added to the *xWebsites* field. If you upload the bundle again, it will have the new web site value appended to the *xWebsites* metadata field, and so forth. The previous value will not be deleted; this means that the site assets you upload this way will have both web sites in the value.

If you upload through the Oracle Content Server, the bundle will not be immediately unpacked. When you upload through Designer, there is a reference point of which web site the bundle should be associated with once unpacked. When uploading through Oracle Content Server, there is no reference for updating the *xWebsites* field. You will be prompted to select which web sites to associate the content with when you unpack the bundle. Unpacking a bundle on the Oracle Content Server is a separate step from uploading a bundle.

When you go to the Manage Archives page (see *Managing Oracle Site Studio*), you will need to further unpack it and associate it with the appropriate web site or sites.

Uploading from Designer

To upload a definition bundle to the Oracle Content Server from Designer, perform these steps:

1. From the menu bar, select **File**, then select **Definitions**, then select **Upload Definition Bundle...**
2. In the Open dialog, browse to the definition bundle on your file system and then click **Open**.
3. The definition bundle is automatically uploaded to the Oracle Content Server.

Once uploaded, it is automatically unpacked and the xWebsites metadata field data is filled with the web site that was open in Designer when you uploaded the bundle.

Working With Content Files

Content files are the files used to store any of the materials viewed on a web site. These files contain the site content modified by templates and definitions, and the site content editable through Oracle Site Studio Contributor, such as contributor data files and images. Native documents are also content files.

This section covers the following topics:

- [Section 10.1, "About Content Files"](#)
- [Section 10.2, "Creating a New Content File"](#)
- [Section 10.3, "Copying a Content File"](#)
- [Section 10.4, "Viewing the Content Information of a Content File"](#)
- [Section 10.5, "Adding a Content File to a Site"](#)
- [Section 10.6, "Removing a Content File from a Site"](#)
- [Section 10.7, "Editing a Contributor Data File"](#)
- [Section 10.8, "Working With Native Documents"](#)
- [Section 10.9, "Working With Images"](#)

10.1 About Content Files

Content files are the files used to store any of the materials viewed on a web site. These files contain the site content modified by templates and definitions, and editable through Contributor, such as contributor data files and images. Native documents are also content files.

This section is about the types of content files and related files as viewed in the Site Assets pane:

- [Section 10.1.1, "Contributor Data Files"](#)
- [Section 10.1.2, "Native Documents"](#)
- [Section 10.1.3, "Conversion Definitions"](#)
- [Section 10.1.4, "Images"](#)

10.1.1 Contributor Data Files

A contributor data file is an XML file that is created and maintained by Oracle Site Studio. It is used to store the content of a web site and checked in to the content server. If a site contributor wants to update the content of a web page, its associated contributor data file is retrieved from the server and the data made available in the

Contributor editor, where the contributor can modify the content. After the contributor is done making changes to the data file, it is checked back in to the content server and the web page is updated to reflect the new content.

For more information, see [Section 3.16, "Contributor Data Files and Native Documents."](#)

10.1.2 Native Documents

A native document is a third-party file that is viewed in Oracle Site Studio and edited using the application originally used to create it (for example, Microsoft Word is used for Word documents). When the file is checked in, it is converted into HTML (using Dynamic Converter) and then made available to the web page.

For more information, see [Section 3.16, "Contributor Data Files and Native Documents."](#)

10.1.3 Conversion Definitions

A conversion definition is used to transform a native document to display inline on a web site. The conversion definition is a rule, template, or combination that defines how a native document is converted to HTML and displayed in the page.

When you add a native document to a web page, the conversion definitions can specify a (Dynamic Converter) template or rule that is used to convert the document into HTML.

10.1.4 Images

You can add, copy, and remove images from your site. The supported image formats are GIF and JPG. Images cannot be added to a web site until they are added to the content server as an asset.

10.2 Creating a New Content File

To create a content file, perform these tasks:

1. In the menu on the Site Assets pane, select the content file type to create.
2. Click the **Create New** icon ([Figure 10–1](#)), select **New**, and then select the specific document type to create.

The Assign Info Form opens for you to check the item into the content server.

Figure 10–1 Create New Icon



3. Enter appropriate values for the Assign Info Form.
For guidelines on naming Oracle Site Studio assets, see [Section 4.2, "Naming Site Assets."](#)
4. When complete, click **Assign Info**.
5. The content file is created.

You may find that the most common method of creating the contributor data files is when you assign content. When you run the Switch Content wizard (see [Section A.76, "Switch Content Wizard"](#)), the wizard can create a contributor data file for you.

10.3 Copying a Content File

You can select a content file from the list to copy, select a content file from the content server to copy, or select a content file from your local instance to copy.

Copying a content file from the Site Assets pane

1. In the menu on the Site Assets pane, select the content file type.
2. Select the content file you want to copy from the list.
3. Click the **Create New** icon (Figure 10–2), select **Copy**, and then **Selected**:
The Assign Info Form opens for you to check the item into the content server.

Figure 10–2 *Create New Icon*



4. Enter appropriate values for the Assign Info Form.
5. When complete, click **Assign Info**.
6. The content file is copied.

Copying a content file from the content server

1. In the menu on the Site Assets pane, select the content file type.
2. Click the **Create New** icon (Figure 10–2), select **Copy**, and then **from Server**.
A search results page opens.
3. Click the corresponding **Select** button of the one you want to copy.
The Assign Info Form opens for you to check the item into the content server.
4. Enter appropriate values for the Assign Info Form.
5. When complete, click **Assign Info**.
6. The content file is copied.

Copying a content file from your local instance

1. In the menu on the Site Assets pane, select **the content file type**.
2. Click the **Create New** icon (Figure 10–2), select **Copy**, and then **from Local**.
A navigation window opens.
3. On your local instance, navigate to the content file you want to copy.
4. Select the contributor data file and click **Open**.
The Assign Info Form opens for you to check the item into the content server.
5. Enter appropriate values for the Assign Info Form.
6. When complete, click **Assign Info**.
7. The content file is copied.

10.4 Viewing the Content Information of a Content File

To view content information for a content file, perform these tasks:

1. In the menu on the Site Assets pane, select the content file type.
2. Select a content file from the list.
3. Click the **Doc Info** icon (Figure 10–3).
The content information page opens.

Figure 10–3 Doc Info icon



10.5 Adding a Content File to a Site

To add a content file to a site, perform these tasks:

1. In the menu on the Site Assets pane, select the content file type.
2. Click the **Add to Site** icon (Figure 10–4).
A search results page opens.

Figure 10–4 Add to Site Icon



3. Select the content files to add, click **Oracle Site Studio**, and then **Select Marked Documents**.
4. Depending on your configuration, you may receive a caution that you are about to add existing asset(s) to your site. Click **OK**.
The content files are now associated with the web site.

10.6 Removing a Content File from a Site

To remove a content file from a site, perform these tasks:

1. In the menu on the Site Assets pane, select the content file type.
2. From the list, select the content file to be removed and click the **Remove From Site** icon (Figure 10–5).

When you remove a content file, you are simply removing its association with a site. You are not deleting the file from the content server.

Figure 10–5 Remove from Site Icon



10.7 Editing a Contributor Data File

To edit the source of a contributor data file, perform these tasks:

1. In the menu on the Site Assets pane, select **Contributor Data Files**.
2. From the list, select the contributor data file to be edited and click the **Edit** icon (Figure 10–6).

The selected contributor data file is opened and can be edited in source view.

Figure 10–6 Edit Icon

The design of a contributor data file is such that editing should be done through Contributor, rather than in Source view in Designer. While it is possible to edit the file directly, it is much easier to edit the contributor data file through Contributor or switch to a different file through the Switch Content Wizard (see [Section 12.3, "Assigning Content With The Switch Content Wizard"](#)).

10.8 Working With Native Documents

In Oracle Site Studio, you can assign a contributor data file or a native document to a contribution region, and individual contributors can add and edit contributor data files and native documents on the web site.

A contributor data file is generated by Oracle Site Studio. It is an XML file that is designed to be edited using the Contributor application. As an XML file, it is in a web-ready format that can be easily pulled into a web page (similar to the way an include file can be pulled into another web page).

A native document, however, is generated by a third-party application (Microsoft Word, Excel, PowerPoint, and so on) and is generally introduced by you or a contributor. The document, in its original state, cannot be viewed on a web page so it is converted into a web page (using Dynamic Converter) when the server delivers the page to the browser.

Unlike contributor data files, native documents do not require the Contributor for editing. The converted document opens in the placeholder as the page opens, and the appropriate application (Microsoft Word, and so on) is loaded when the contributor edits the data.

This section covers the following topics:

- [Section 10.8.1, "Requirements for Native Documents"](#)
- [Section 10.8.2, "Editing a Native Document"](#)
- [Section 10.8.3, "Using a Conversion Definition With Native Documents"](#)
- [Section 10.8.4, "Using Dynamic Converter Template Selection Rules"](#)
- [Section 10.8.5, "Sample Fragments for Native Documents"](#)
- [Section 10.8.6, "Using Direct Delivery"](#)

10.8.1 Requirements for Native Documents

A native document can be assigned to a contribution region in the same way that a contributor data file can. Native documents are added to the site in the same way that contributor data files are added to the site (create a link to the file, add the file to a dynamic list, or check the file in so that its metadata matches a dynamic list or a query performed on the site).

Unlike contributor data files, however, native documents require additional configuration steps on the server so that they appear on the web site as intended. These steps include the following:

- You must have Dynamic Converter installed and running on the content server so that it can convert native documents into web pages.

- You must have Check Out and Open installed and running on the content server so that contributors can check out and edit a native document directly from the web site using the menu on the contribution graphic.
- The native documents must be a supported file format of Dynamic Converter.
- By default, a blank Dynamic Converter template (one with minimal settings) is used to convert the document so that it looks similar to the original. To further customize the appearance of the web page so that it matches the site, you should fine tune the template (or templates) used to convert the document (see [Section 10.8.3, "Using a Conversion Definition With Native Documents"](#)).
- You, or the contributors, must have the associated third-party application installed on the computer that is being used to edit the file.
- If you, or contributors, plan to add native documents to the site using a method outside of Oracle Site Studio (for example, using the Oracle Content Server check-in form, WebDAV Client, or Web Folders), then you must check those files in with the same metadata used by the site, and if necessary, a section of the site.

10.8.2 Editing a Native Document

To edit a native document, perform these tasks:

1. In the menu on the Site Assets pane, select **Native Documents**.
2. From the list, select the native document to be edited and click the **Edit** icon ([Figure 10–7](#)).

You must have Check Out and Open installed to edit a native document. Documents are opened in the appropriate application. For example, Microsoft Word or PowerPoint.

Figure 10–7 Edit Icon



3. When you complete your edits, save the file. It is saved to the server.

10.8.3 Using a Conversion Definition With Native Documents

When you assign a native document to a contribution region (or allow contributor to add native documents to the site), conversions are used to transform native documents to display inline on a web site. Conversion definitions specify the rules that are used to control how a native document is converted to HTML and displayed in the page.

When you have a defined conversion definition, adding native documents to the web page is as simple as adding any other asset.

To add a conversion definition to a region template, follow these steps:

1. In the region template, right-click. The contextual menu opens.
2. Choose **Insert Object**, then **Dynamic Conversion**.
The Insert Dynamic Conversion dialog (see [Section A.78, "Insert Dynamic Conversion Dialog"](#)) appears.
3. Select the conversion definition or enter the name of a new conversion definition.
4. Click **OK**.

5. The conversion definition is now part of the region template. A `wcmDynamicConversion` tag (see *Technical Reference for Oracle Site Studio*) is written to the Region Template and the name of the conversion definition is a parameter in the tag. You can now assign a native document by using the Assign Content dialog (see [Section A.54, "Assign Content Dialog"](#)), or the contributor can add the native document.

The `wcmConversionDefinition` tag can also be written in source view directly.

With native documents, you do not have the kind of formatting control over the final web page like you do with contributor data files (which can be precisely controlled in an element).

As such, you should spend time designing conversion templates specific to your site and then encouraging contributors to use styles in the native document that you can map to styles in the conversion template.

10.8.4 Using Dynamic Converter Template Selection Rules

When you have native documents on your site, one thing you must do is specify a conversion definition to control the look and feel of those documents on the web site. You can specify conversion templates or rules (which govern the use of conversion templates) in the conversion definition by using the Native Document Conversion Settings dialog (see [Section A.25, "Native Document Conversion Settings Dialog"](#)).

The available templates and rules that appear in this dialog must be set up ahead of time using the Dynamic Converter Template Selection Rules page. To access the Template Selection Rules page you must have the dynamic converter component installed on your content server. From the main menu of the server, select **Administration** then **Dynamic Converter Admin** and then **Template Selection Rules**.

Figure 10–8 Dynamic Converter Template Selection Rules Page

Template Selection Rules

Template Selection Rules (In order of evaluation)

Move Up
Move Down
Delete Rule

New rule name:
Add New Rule

Criteria for selected rule

Field Value And

Template and layout for selected rule

Template Available Templates Template Types
Edit Template GUI Template

Layout Available Layouts

File Extension

File extension of the dynamic converted page.

Forced Conversion
Indicates that this rule is to be used for forced conversions. This rule will always be applied to content items as long as the content items match the rule criteria. The conversion results can be retrieved using the `incDynamicConversionByRule` Idoc Script function or the `GET_DYNAMIC_CONVERSION` service with the `conversionRule` parameter specified.

Exclude From User Request
Use this option to prevent the rule to be used when a user clicks on the HTML Rendition link or menu item. Rules designed for fragments and used by the `incDynamicConversionByRule` Idoc Script function should be excluded from Dynamic Converter's rule evaluation during a user request.

Update Quick Help

On the Template Selection Rules page, you can create and edit rules, layouts, and more. For more information, see the *Administrator's Guide for Dynamic Converter*.

10.8.5 Sample Fragments for Native Documents

In the Toolbox, under "Other Fragments," there are a few fragments designed specifically for native documents that you can use. Each fragment enables you to add a native document directly to the page template (outside of a placeholder) and perform a dynamic conversion using a conversion template and layout from Dynamic Converter.

- **Dynamic Conversion:** Adds a converted document to your page template. It includes three parameters that you can use to specify the document, a conversion template, and a conversion layout.
- **Dynamic Conversion IFrame:** Similar to the first fragment except that it adds the converted document in an IFrame (so that you can scroll through the entire document without scrolling down the web page). It includes several parameters that you can use to specify the document, the template, and the attributes of the IFrame.
- **Random Conversion.** Performs a random dynamic conversion of native documents based on a query specified in the content server. It includes four

parameters that you can use to specify a query, a query limit, and the conversion templates.

While fragments offer many methods of incorporating native documents, you may find it more useful to directly assign the native document to placeholders. For more information on these fragments, see [Section C.4, "Other Fragments."](#)

10.8.6 Using Direct Delivery

Normally when a native doc is included in a website it is converted to HTML by DC and included in a web page. There are situations where it might be desirable to deliver the file, such as a Microsoft Word document, in its native format instead. Direct delivery allows this option. Using direct delivery, a Microsoft Word document may be delivered as a Microsoft Word document from the same friendly URL that would have previously delivered converted content.

In previous versions of Oracle Site Studio it was possible to deliver PDF files directly; however, this involved redirecting to the weblayout URL and losing the site context. Direct delivery can now be used with PDF and any other native format to deliver the file from the friendly URL without the redirect to the weblayout, thus helping to maintain the users context within the site. This means that a path containing an entry such as `/groups/public/documents` would be seen in the users browser. The feature allows a global list of file extensions to be configured and then overrides may be applied, via custom section properties, at the section level.

For more information on enabling direct delivery, see *Technical Reference for Oracle Site Studio*.

10.9 Working With Images

Images are just as important as text in the construction of a web site. They can serve as displays of information, anchors for text, or simple decoration to improve the look of a web site.

Depending on the content server configuration, multiple renditions of an image may be automatically created and stored on the server. For instance, a small preview version of the image or a lower resolution of the image that was uploaded to the content server.

Working With Scripts and Forms

In Oracle Site Studio, you can apply a custom form or script to an element to enhance the contributor's experience beyond the out-of-the-box behavior.

This section covers the following topics:

- [Section 11.1, "About Scripts and Forms"](#)
- [Section 11.2, "Managing Assets"](#)
- [Section 11.3, "Custom Element Forms"](#)
- [Section 11.4, "Custom Configuration Scripts"](#)
- [Section 11.5, "Default Validation Options"](#)

11.1 About Scripts and Forms

To enhance the contributor's experience beyond the out-of-the-box behavior in an element, you can use a custom form or script that you apply to an element.

You do this to allow contributors to add objects that are not supported in a WYSIWYG element, as an example, or as a mechanism that the designer uses to enforce content requirements or restrictions for the contributor.

Oracle Site Studio includes several sample forms and scripts in the content server (described in the topics above) that you can begin using right away. To help you organize, edit, and manage these files, you may want to add each one to your site as a site asset.

This section is about these topics:

- [Section 11.1.1, "About Scripts"](#)
- [Section 11.1.2, "About Custom Configuration Scripts"](#)
- [Section 11.1.3, "About Custom Element Forms"](#)
- [Section 11.1.4, "About Validation Scripts"](#)

11.1.1 About Scripts

Scripts are often used in a web site to make the web site more effective in displaying information or in interacting with the user. You can add scripts, edit scripts, copy scripts, and add or remove scripts from your site.

11.1.2 About Custom Configuration Scripts

Custom Configuration Scripts are used to modify the interface for editing within an element in Contributor. Each configuration script can be used to add or replace commands, toolbar buttons, and context menu items. Any part of the editor can be customized to fulfill a specific requirement.

For more information, see [Section 11.1.2, "About Custom Configuration Scripts."](#)

11.1.3 About Custom Element Forms

A custom element form is designed to work with a custom element definition, which you add to a region template and region definition like any other element. The custom element form is used to create a custom interface that contributors use to add and control various types of content to a web site. Oracle Site Studio comes with several predefined custom element forms (in `[CS-Dir]\custom\SiteStudio\support`). Sample forms are also checked into the Oracle Content Server during installation.

11.1.4 About Validation Scripts

You can enforce site rules such as what content is added using your own validation scripts. Validation scripts can only be written in JavaScript.

Validation scripts are associated with element definitions, and are used to validate the edits that contributors make. Some things that validation scripts could be used for include size requirements on images, limiting text to 140 characters, and so forth.

11.2 Managing Assets

This section covers the following topics:

- [Section 11.2.1, "Creating a New Script"](#)
- [Section 11.2.2, "Copying a Script"](#)
- [Section 11.2.3, "Editing a Script"](#)
- [Section 11.2.4, "Viewing the Content Information of a Script"](#)
- [Section 11.2.5, "Adding a Script to a Site"](#)
- [Section 11.2.6, "Removing a Script from a Site"](#)

11.2.1 Creating a New Script

To create a script, perform these tasks:

1. In the menu on the Site Assets pane, select **Scripts**.
2. Click the **Create New** icon ([Figure 11-1](#)), select **New**, and then **Javascript File**.

The Assign Info Form opens for you to check the item into the content server.

Note: VBScript cannot be used to write or run custom validation scripts.

Figure 11-1 Create New Icon



3. Enter appropriate values for the Assign Info Form.
For guidelines on naming Oracle Site Studio assets, see [Section 4.2, "Naming Site Assets."](#)
4. When complete, click **Assign Info**.
5. The script is created.

When a new script is added, there is no code in it. It is blank.

Note: When you create a new script by selecting Validation Scripts (rather than Scripts) using the Assets pane, the customization is pre-populated with code compatible with an FCK element.

11.2.2 Copying a Script

You can select a script from the list to copy, select a script from the content server to copy, or select a script from your local instance to copy.

Copying a script from the list

1. In the menu on the Site Assets pane, select **Scripts**.
2. Select the script you want to copy from the list.
3. Click the **Create New** icon ([Figure 11-2](#)), select **Copy**, and then **Selected**:
The Assign Info Form opens for you to check the item into the content server.

Figure 11-2 Create New Icon



4. Enter appropriate values for the Assign Info Form.
5. When complete, click **Assign Info**.
6. The script is copied.

Copying a script from the server

1. In the menu on the Site Assets pane, select **Scripts**.
2. Click the **Create New** icon ([Figure 11-2](#)), select **Copy**, and then **from Server**.
A search results page opens.
3. Click the corresponding **Select** button of the one you want to copy.
The Assign Info Form opens for you to check the item into the content server.
4. Enter appropriate values for the Assign Info Form.
5. When complete, click **Assign Info**.
6. The script is copied.

Copying a script from your local instance

1. In the menu on the Site Assets pane, select **Scripts**.
2. Click the **Create New** icon ([Figure 11-2](#)), select **Copy**, and then **from Local**.
A navigation window opens.

3. On your local instance, navigate to the script you want to copy.
4. Select the script and click **Open**.
The Assign Info Form opens for you to check the item into the content server.
5. Enter appropriate values for the Assign Info Form.
6. When complete, click **Assign Info**.
7. The script is copied.

11.2.3 Editing a Script

To edit a script, perform these tasks:

1. In the menu on the Site Assets pane, select **Scripts**.
2. From the list, select the script to be edited and click the **Edit** icon (Figure 11-3).
The selected script is opened and can be edited in source view.

Figure 11-3 Edit Icon



11.2.4 Viewing the Content Information of a Script

To view content information for a script, perform these tasks:

1. In the menu on the Site Assets pane, select **Scripts**.
2. Select a script from the list.
3. Click the **Doc Info** icon (Figure 11-4).

The content information page opens.

Figure 11-4 Doc Info icon



11.2.5 Adding a Script to a Site

To add a script to a site, perform these tasks:

1. In the menu on the Site Assets pane, select **Scripts**.
2. Click the **Add to Site** icon (Figure 11-5).

A search results page opens.

Figure 11-5 Add to Site Icon



3. Select the scripts to add, click **Oracle Site Studio**, and then **Select Marked Documents**.
4. Depending on your configuration, you may receive a caution that you are about to add existing asset(s) to your site. Click **OK**.

The scripts are now associated with the web site, and you can now open it to edit.

11.2.6 Removing a Script from a Site

To remove a script from a site, perform these tasks:

1. In the menu on the Site Assets pane, select **Scripts**.
2. From the list, select the script to be removed and click the **Remove From Site** icon (Figure 11–6).

When you remove a script, you are simply removing it from the site, not deleting from the content server.

Figure 11–6 *Remove from Site Icon*



11.3 Custom Element Forms

A custom element is a user-defined element. In addition to the other elements (WYSIWYG, text only, image only, static list, dynamic list), the custom element provides one way to extend Oracle Site Studio to suit individual business needs. Custom elements are created when the other elements are not as suitable.

Custom elements are full HTML files that reside within an IFRAME in the Contributor form. A custom element must use an API and implement several callbacks to function correctly as an Oracle Site Studio element.

Important: If you upgrade an Oracle Site Studio site from a version before 10gR3 (10.1.3.3.3), you must manually update any custom elements in the site. For more information, see [Section 11.3.2, "Backward Compatibility and Upgrading."](#)

A custom element form is designed to work with a custom element, which you add to a contribution region like any other element. The custom element form is unique in that it can be used to create a custom interface that contributors use to add various types of content to a web site.

You might, for example, create a custom form that queries the content server for a certain file type, enables users to select one of those files, and returns the user to the Contributor. Or, you might create a form that provides a text box that contributors use to copy and paste source code or formatted text to a web page.

The form can be a generic browser-based form that collects arbitrary information from the user and returns a value to the host (in this case, Contributor).

This section includes the following topics:

- [Section 11.3.1, "Implementing a Custom Element"](#)
- [Section 11.3.2, "Backward Compatibility and Upgrading"](#)
- [Section 11.3.3, "Sample Custom Elements"](#)

11.3.1 Implementing a Custom Element

In order for a custom element to function properly in a Contributor form, a custom element must use an API and implement a hand-full of callbacks. The following notes highlight the ElementAPI and its methods for creating custom elements.

Note: The path (URL) to a custom element must be in the same domain as the Contributor form. This is so that the Contributor form and the custom element can communicate without violating cross-domain scripting rules.

ElementAPI

The ElementAPI object is a JavaScript object explicitly loaded into the custom element page that facilitates communication between the Contributor form and the custom element. The ElementAPI provides methods for custom elements to communicate to the Contributor form, and a callback mechanism for the Contributor form to pass notifications to the custom element.

Loading the ElementAPI JavaScript Object

Before the ElementAPI and its supporting libraries can be used, the ElementAPI must first be loaded into the custom element page. After the ElementAPI is loaded, the custom element should continue with page initialization and notify the Contributor form that the custom element is loaded and ready to go.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Simple Custom Element</title>

    <style type="text/css">
      * { padding: 0; margin: 0; }
      body { font-size: 10pt; font-family: Tahoma; }
      #container { height: 150px; overflow: auto; width: 100%; text-align: center;
align: center;}
      #content { width: 99%; }
    </style>

    <script type="text/javascript">
      var Custom = {};

      Custom.originalData = null;

      Custom.Initialize = function()
      {
        // Set callback methods for the Contributor Form to send notifications to this
        Element.
        ElementAPI.SetCallback('GetElementContent', function(callback)
        {
          callback(Custom.GetData());
        });
        ElementAPI.SetCallback('ActivateElement', function(callback){
$ID('content').focus(); callback(); });
        ElementAPI.SetCallback('CanCloseElement', function(callback){
callback({canClose: true}); });
        ElementAPI.SetCallback('Show', function(callback){ callback(); });
        ElementAPI.SetCallback('Hide', function(callback){ callback(); });
        ElementAPI.SetCallback('IsDirty', function()
        {
          return { isDirty: (Custom.originalData !== Custom.GetData()) };
        });
      };
```

```

// Set the height of the Custom Element form's host.
var height = ( ElementAPI.GetElementCount() == 1 ) ? '100%' : '150px';
var flags = ElementAPI.GetElementConfigProperty('flags') || {};
var config = ElementAPI.GetElementConfiguration() || {};
if( flags.canSetHeight && config.height && ( config.height.length > 0 ) )
{
    height = config.height;
}
height = ElementAPI.SetHostHeight(height);

// Get the default content stored within the data file.
$ID('content').value = Custom.GetDefaultData();

// Add base styles
WCM.DHTML.AddStyleSheet({path: WCM.path + './base/wcm.base.css', context:
window});

// Add a resize handler for when the window is resized.
var ResizeHandler = function()
{
    WCM.DHTML.SetStyle('container', 'height', WCM.DHTML.GetViewportHeight() +
'px');
    WCM.DHTML.SetStyle('container', 'width', WCM.DHTML.GetViewportWidth() + 'px');
};
ResizeHandler();

WCM.DHTML.AddWindowResizeEvent(window, ResizeHandler);

// Immediately store the original data as a baseline for the IsDirty
comparison.
Custom.originalData = Custom.GetData();

// Let the Contributor Form know this Custom Element is ready. (required)
ElementAPI.Ready();
};

Custom.GetDefaultData = function()
{
    return ElementAPI.GetDefaultData()
        .replace(/<pre>/g, '')
        .replace(/<\n/pre>/g, '')
        .replace(/&lt;/g, "<")
        .replace(/&gt;/g, ">") || 'Some default content.';
};

Custom.GetData = function()
{
    return '<pre>' + $ID('content').value.replace(/</g, "&lt;").replace(/>/g,
"&gt;") + '</pre>';
};

try {
    // Tell the Contributor Form to load the Custom Element dependent JavaScript
libraries. (required)
    // PARAMETERS
    // 1st parameter: The context (window) in which the form loads the Custom
Element dependent JavaScript libraries into.
    // 2nd parameter: A user defined function pointer for the Contributor Form to
call when initialization is complete.
    window.top.WCM.InitializeCustomElement(window, Custom.Initialize);
}

```

```

        catch(e) { }
    </script>

</head>
<body>
  <div id="container">
    <div id="title">Simple Custom Element</div>
    <div><textarea id="content" rows="5"></textarea></div>
  </div>
</body>
</html>

```

Note: The code example loads the ElementAPI and notifies the Contributor form that it is loaded and ready. It does not, however, collect or save data.

Communication from Contributor Form to Custom Element

The Contributor form communicates with a custom element by executing functions implemented by the custom element. As part of the initialization process, a custom element must register these functions by passing their function pointers to the Contributor form.

The table below lists the functions that can be registered with the Contributor form. None of these functions must be implemented by the custom element; however, a few of them are required if the intention is to collect and save data from a Contributor user. Furthermore, all of these functions (except the `IsDirty()` function), when executed, pass a callback function pointer to execute when the task is complete. This enables asynchronous communication if a custom element must perform an asynchronous task during execution.

Function Signature	Description
<code>CanCloseElement(callback);</code>	The Contributor form executes this method when the Contributor user performs an update. The implementation of the function should calculate whether the custom element can be safely closed. For instance, if the data does not pass validation, then the custom element should indicate that it cannot be closed.
<code>GetElementContent(callback);</code>	The Contributor form executes this method when the Contributor user performs an update. The implementation of the function should pass back string content to be saved.
<code>Hide(callback);</code>	The Contributor form executes this method whenever the form performs a DHTML task that overlays an HTML element over the custom element. For instance, this method is executed when the Metadata tab is activated and the Contributor elements are obscured. This method was introduced specifically for the Ephox-based elements, because Java applets always have top z-index. All other elements (HTML-based elements) can ignore this method.

Function Signature	Description
Show(callback);	The Contributor form executes this method whenever the form performs a DHTML task that removes an overlay that makes custom element reappear. This method was introduced specifically for the Ephox-based elements because Java applets always have top z-index. All other elements (HTML-based elements) can ignore this method.
IsDirty();	The Contributor form executes this method whenever the form popup is being closed without updating. The custom element should calculate whether unsaved changes exist so that the Contributor user can be notified if there are unsaved changes.

The following is a JavaScript code snippet of how a custom element can register functions with the Contributor form:

```
function CanCloseElement(callback)
{
    // No data validation in this sample - just pass back a true value.
    callback({canClose: true});

    // Here is an example of passing a false value
    // callback({canClose: false, reason: 'Failed validation. Use only lowercase
    // letters.'});
}

function GetElementContent(callback)
{
    // Pass back some sample content for demo purposes.
    callback('This is my Custom Element Content.');
```

```
function Show(callback)
{
    // Just handle this notification by executing the callback.
    callback();
}
```

```
function Hide(callback)
{
    // Just handle this notification by executing the callback.
    callback();
}
```

```
function IsDirty()
{
    // This Custom Element is never dirty - so pass a false value.
    return {isDirty: false};
}
```

```
// Set callback methods for the Contributor Form to send notifications to this
// Element.
ElementAPI.SetCallback('CanCloseElement', CanCloseElement);
ElementAPI.SetCallback('GetElementContent', GetElementContent);
ElementAPI.SetCallback('Show', Show);
ElementAPI.SetCallback('Hide', Hide);
ElementAPI.SetCallback('IsDirty', IsDirty);
```

Communication From Custom Element to Contributor Form

A custom element initiates communication with the Contributor form by using the ElementAPI JavaScript object. The following is a list of available ElementAPI methods.

Function Signature	Description
ElementAPI.GetDefaultData();	Retrieves the default content stored in the data file.
ElementAPI.GetSearchResults(options);	Opens the Oracle Content Server's Get Search Results page.
ElementAPI.GetQueryText(options);	Opens the Get Query Text UI.
ElementAPI.CaptureQuery(options);	Opens the Oracle Content Server's Capture Query page.
ElementAPI.GetHyperlink(options);	Opens the Hyperlink Wizard UI.
ElementAPI.FocusForm(options);	Focuses the parent window thereby blurring the Element window.
ElementAPI.SetHostHeight(height);	Sets the height of the Element's containing IFRAME.
ElementAPI.SetRequiredIndicator(isRequired);	Toggles the Required graphic indicator in the Contributor Form UI.
ElementAPI.GetSite(options);	Opens the Choose Website picker UI.
ElementAPI.GetSection(options);	Opens the Choose Website Section picker UI.
ElementAPI.GetColor(options);	Opens the Color picker UI.
ElementAPI.GetFont(options);	Opens the Get Font picker UI.

ElementAPI Dependent Scripts

When the ElementAPI is loaded into the custom element page, so are the ElementAPI dependent scripts. These scripts contain most of the JavaScript WCM library and is also available for custom element authors to use. The following script files are loaded into a custom element:

- wcm.js
- ./base/wcm.dhtml.js
- ./base/wcm.get.js
- ./base/wcm.http.js
- ./base/wcm.popup.js
- ./sitestudio/wcm.contentserver.popup.js
- ./form/elements/wcm.elementapi.js
- ./sitestudio/elements/wcm.sitestudio.elementapi.js
- ./sitestudio/wcm.idc.js
- ./form/elements/element/wcm.element.js
- ./form/elements/custom/wcm.custom.js

Example of a Source-Mode Type of Custom Element

The following simple custom element example saves content entered into a TEXTAREA by Contributor users, and opens it directly in the layout.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Simple Custom Element</title>

    <style type="text/css">
      * { padding: 0; margin: 0; }
      body { font-size: 10pt; font-family: Tahoma; }
      #container
      {
        height: 150px;
        overflow: auto;
        width: 100%;
        text-align: left;
        align: left;
        padding: 7px;
      }
      .controls
      {
        border: 1px solid #B9B9B4;
        border-top-color: #DDDD8;
        border-bottom-color: #797975;
        vertical-align: middle;
        font-family: tahoma, verdana, arial, helvetica, sans-serif;
        font-size: 100%; /* 11px @ Normal */
        font-color: #000;
      }
    </style>

    <script type="text/javascript">
      var Custom = {};

      Custom.originalData = null;

      Custom.Initialize = function()
      {
        ElementAPI.SetCallback('GetElementContent', function(callback)
        {
          callback(Custom.GetData());
        });
        ElementAPI.SetCallback('ActivateElement', function(callback){
          $ID('content').focus(); callback(); });
        ElementAPI.SetCallback('CanCloseElement', function(callback){
          callback({canClose: true}); });
        ElementAPI.SetCallback('Show', function(callback){ callback(); });
        ElementAPI.SetCallback('Hide', function(callback){ callback(); });
        ElementAPI.SetCallback('IsDirty', function()
        {
          return { isDirty: (Custom.originalData !== Custom.GetData()) };
        });

        var height = ( ElementAPI.GetElementCount() == 1 ) ? '100%' : '150px';
        var flags = ElementAPI.GetElementConfigProperty('flags') || {};
        var config = ElementAPI.GetElementConfiguration() || {};
        if( flags.canSetHeight && config.height && ( config.height.length > 0 ) )
        {
          height = config.height;
        }
      }
    </script>
  </head>
  <body>
    <div id="container">
      <div id="controls">
        <input type="text" value="Content" />
      </div>
    </div>
  </body>
</html>

```

```

height = ElementAPI.SetHostHeight(height);

$ID('input-value').value = ElementAPI.GetDefaultData();

WCM.DHTML.AddStyleSheet({path: WCM.path + './base/wcm.base.css', context:
window});

var ResizeHandler = function()

{
var viewPortWidth = WCM.DHTML.GetViewportWidth();
var viewPortHeight = WCM.DHTML.GetViewportHeight();
WCM.DHTML.SetStyle('container', 'height', viewPortHeight + 'px');
WCM.DHTML.SetStyle('container', 'width', viewPortWidth + 'px');
WCM.DHTML.SetStyle('input-value', 'height', (viewPortHeight - 15) + 'px');
WCM.DHTML.SetStyle('input-value', 'width', (viewPortWidth - (WCM.IS_IE ? 15
: 15)) + 'px');
};
ResizeHandler();

WCM.DHTML.AddWindowResizeEvent(window, ResizeHandler);
Custom.originalData = Custom.GetData();
ElementAPI.Ready();
};

Custom.GetData = function()
{
return $ID('input-value').value;
};

try {
window.top.WCM.InitializeCustomElement(window, Custom.Initialize);
} catch(e) { }
</script>

</head>
<body>
<div id="container" nowrap>
<textarea id="input-value" class="controls" type="text" title="URL"></textarea>
</div>
</body>
</html>

```

11.3.2 Backward Compatibility and Upgrading

All custom element forms compatible with Oracle Site Studio releases before Oracle Site Studio 10gR3 (10.1.3.3.3) are not compatible with Oracle Site Studio 10gR4 (10.1.4) and later releases, and they must be manually upgraded (re-authored). The primary reason for not maintaining backward compatibility is due to Oracle Site Studio's prior dependency upon Internet Explorer's proprietary 'window.external' functionality. This is because Contributor used an ActiveX control. The window.external functionality blocked at the point of code execution and is not easily duplicated in a cross-browser/platform DHTML solution. The upside to breaking backward compatibility is that the new custom elements are much more flexible and better integrated into the Contributor application architecture (in addition to being a cross-browser/platform solution).

Detecting Legacy Custom Element Forms

A legacy custom element form (that is, pre-10gR3, 10.1.3.3.3), if loaded into the new Contributor application, is detected (by default) and an error message opens in its place within the Contributor form. The Contributor application does this by first downloading the custom element form, parsing the source code, and ascertaining whether the custom element form is compatible with the new Contributor application.

The functionality and overhead to detect legacy custom element forms is unnecessary on production installations and is turned off by default. If you have legacy custom element forms to upgrade, you need to turn the flag on. To turn on legacy custom element form detection, add the following line to the Oracle Content Server's *config.cfg* file and restart the server:

```
SSValidateCustomElements=true
```

11.3.3 Sample Custom Elements

Oracle Site Studio 11gR1 comes with eleven sample custom forms or elements. These sample forms are example implementations of custom elements that work within the Contributor form framework.

These samples are checked into the content server with the *xWebsiteObjectType* metadata value of 'Custom Element Form' upon installation of the Oracle Site Studio component.

ss_boolean_checkbox_form.htm

This sample form is an example of a simple custom element with a checkbox.

ss_boolean_radio_form.htm

This sample form is an example of a simple custom element with a radio button.

ss_docname_form.htm

This sample form is an example of a custom element that enables a content server search to select a Content ID.

ss_option_form.htm

This sample form is an example of a custom element that enables you to build an option list. This example is a very simple example. In more complex instances, the list could be populated from many other sources, such as content server metadata.

ss_query_form.htm

This sample form is an example of a custom element that enables you to capture a content server query.

ss_url_form.htm

This sample form is an example of a custom element that enables a content server search to return a weblayout URL.

ss_textarea_form.htm

This sample form is an example of a simple custom element with minimal amount of code. This particular example implements a text-only custom element utilizing a TEXTAREA HTML element, which is a very lightweight method of providing a text-only editor.

ss_integer_form.htm

This sample form is an example of a custom element for capturing an integer value. This is useful for capturing a parameter to use elsewhere.

ss_simple_form.htm

This sample form is an example of a simple custom element with minimal amount of code. This particular example implements a text-only custom element utilizing a TEXTAREA HTML element, which is a very lightweight method of providing a text-only editor.

ss_flash_form.htm

This sample form is an example of a custom element that enables Contributor users to select a FLASH object from the content server.

ss_sample_form.htm

This sample form is an example of a custom element that uses all available ElementAPI methods. A complete list of these methods is available in [Section 11.3.1, "Implementing a Custom Element."](#)

11.4 Custom Configuration Scripts

Custom configuration scripts are javascript files that are used specifically to modify the contributor's editing area for each individual element. A configuration script can be associated with different elements, but each element must individually call the script.

The customizations affect the FCKeditor and Ephox-based elements, which are WYSIWYG, text only, and image only. Custom configuration scripts can be used to add or replace commands, toolbar buttons, and context menu items. Any part of the editor can be customized to fulfill a specific requirement.

Oracle Site Studio ships with several sample custom configuration scripts. These samples are checked into the content server with the *xWebsiteObjectType* metadata value of "Custom Configuration Script" when the Oracle Site Studio component is installed on the content server.

The following scripts are the included samples:

FCKeditor-Compatible Sample Scripts

- `ss_fck_simple_command.js`: This sample script is an example of a simple FCKeditor command. This particular example implements an FCKeditor command, toolbar button, and menu item.
- `ss_fck_image_command.js`: This sample script is an example of an FCKeditor command that overrides the existing "ssImage" and "Image" commands. In this example, the commands are redefined and the existing Image toolbar button and context menu items are re-used.
- `ss_fck_load_multiple_commands.js`: This sample script shows how to load multiple Custom Configuration Scripts.
- `ss_fck_template_command.js`: This sample script enables users to insert pre-defined HTML.

For more information on the Custom Configuration Scripts in the Site Assets pane, see [Chapter 11, "Working With Scripts and Forms."](#)

11.5 Default Validation Options

Depending on the type of element you're working with, you see different default validation options in the Element Validation dialog (see below):

Element type	Validation options
WYSIWYG	<p>min: Enforces a minimum amount of text (in characters).</p> <p>max: Enforces a maximum amount of text (in characters).</p> <p>Disallowed characters: Prevents contributors from using the characters that you specify here.</p> <p>Allow line breaks
: When checked, contributors can create a line break (soft return), which is usually done by pressing SHIFT + ENTER on the keyboard.</p> <p>Allow paragraphs <p>: When checked, contributors can create a new paragraph (hard return), which is usually done by pressing ENTER on the keyboard.</p> <p>Allow images : When checked, contributors can add images.</p>
Plain Text	<p>min: Enforces a minimum amount of text (in characters).</p> <p>max: Enforces a maximum amount of text (in characters).</p> <p>Disallowed characters: Prevents contributors from using the characters that you specify here.</p> <p>Allow multiple lines: When checked, contributors can add new lines of text (by pressing ENTER on their keyboard).</p>
Image	<p>Width:</p> <ul style="list-style-type: none"> ▪ min: Enforces a minimum width (in pixels). ▪ max: Enforces a maximum width (in pixels). <p>Height:</p> <ul style="list-style-type: none"> ▪ min: Enforces a minimum height (in pixels). ▪ max: Enforces a maximum height (in pixels).
Static List	<p>Number Rows</p> <ul style="list-style-type: none"> ▪ min: Enforces a minimum number of rows. ▪ max: Enforces a maximum number of rows.

The first thing you do in Oracle Site Studio is connect to a content server and create your assets. You do this using the Connection Manager in Oracle Site Studio Designer. With the Connection Manager, you identify a content server, create a web site on that server (more specifically, a project file), and create a connection name that is used by Oracle Site Studio Designer to connect to the site.

As soon as you've done this, you can start constructing your site assets into a web site, using a site hierarchy to define the arrangement of the information. A site hierarchy comprises the individual sections of your site, like *Products*, *Services*, and *About Us*. In each section, there is typically a primary page, used as the main page for that section. A secondary page is usually used as well.

You spend a considerable amount of time setting up the site and site hierarchy, which is discussed next.

- [Section 12.1, "Working With the Site Hierarchy"](#)
- [Section 12.2, "Working With Fragments"](#)
- [Section 12.3, "Assigning Content With The Switch Content Wizard"](#)
- [Section 12.4, "Working With Links"](#)
- [Section 12.5, "Comparing Changes on a Web Page"](#)

12.1 Working With the Site Hierarchy

A site hierarchy is a collection of individual sections and web pages that comprise the structure of the site. Similar to a folder in a folder hierarchy, a section in a site hierarchy is used to store related information. The site hierarchy can be flat, where all sections are located at the top, or it can be deep, where numerous sections are placed within other sections.

While you use the site hierarchy to organize and categorize content on your web site, Oracle Site Studio uses the site hierarchy to perform numerous "behind-the-scenes" tasks, like generating navigational links and organizing page templates.

Site navigation: The site hierarchy becomes the basis for your overall site navigation. Oracle Site Studio "reads" the site hierarchy and converts each section into a link, which you see when you use a navigation fragment (see [Section C.1, "Navigation Fragments"](#)). If you have sections called *Products*, *Services*, and *Contact Us* in the site hierarchy, those names appear in the navigation menu for users to click.

Page templates: The site hierarchy is also used to organize content (contributor data files, native documents, fragments, and so on) to display on a page template. Rather than create numerous web pages in a section of the web site, you create just one

primary page and one secondary page. Oracle Site Studio assembles content on either the primary page or secondary page, depending on how you organize your site hierarchy.

You generally perform these tasks with your site hierarchy:

- [Section 12.1.1, "Adding a Section to the Site Hierarchy"](#)
- [Section 12.1.2, "Renaming a Section in the Site Hierarchy"](#)
- [Section 12.1.3, "Rearranging Sections in the Site Hierarchy"](#)
- [Section 12.1.4, "Removing a Section From the Site Hierarchy"](#)
- [Section 12.1.5, "Assigning a Home Page"](#)
- [Section 12.1.6, "Assigning a Primary Page"](#)
- [Section 12.1.7, "Assigning a Secondary Page"](#)
- [Section 12.1.8, "Showing Sections in the Site Navigation"](#)
- [Section 12.1.9, "Designating a Section as an Error Handler"](#)
- [Section 12.1.10, "Changing the Security of the Site or a Section"](#)
- [Section 12.1.11, "Viewing and Editing Web Site and Section Properties"](#)
- [Section 12.1.12, "Defining Custom Properties for a Section"](#)
- [Section 12.1.13, "Expanding and Collapsing the Site Hierarchy"](#)
- [Section 12.1.14, "Refreshing Your Site Hierarchy in Designer"](#)

You can delegate many of these site hierarchy tasks to a site manager by enabling the Manager application (see [Chapter 16, "Setting Up Manager"](#)).

12.1.1 Adding a Section to the Site Hierarchy

You add a section to your web site using the Site Hierarchy pane in Designer.

To add a section, perform these tasks:

1. In the Site Hierarchy pane, right-click your web site and choose **Add New Section**. This opens the New Section dialog (see [Section A.8, "New Section Dialog"](#)).
2. Enter a value for **Label**. The Label appears in the Site Hierarchy pane and in the navigation menus on your site.
3. Enter a value for **URL**. The URL is used in the site address and all (path-based) links that target this section.

Note: Oracle Site Studio automatically inserts a value that contains no spaces or special characters.

4. Choose how you would like this section to be identified. The section ID, also referred to as a node ID, is used in all (ID-based) links that target this section.
 - Choose **Auto-generate section ID** to have an ID automatically generated by Oracle Site Studio.
 - Choose **Manually enter section ID** to specify your own ID for the section. Enter your ID in the space provided.

Important: The section ID must be unique in the web site.

5. Click **OK** to close this dialog and return to the site hierarchy.

To make the new section appear in the navigation menus on your site (generated by navigation fragments), see [Section 12.1.8, "Showing Sections in the Site Navigation."](#)

To add a subsection (a section within a section), right-click an existing section in the site hierarchy and choose **Add New Section**.

12.1.2 Renaming a Section in the Site Hierarchy

You can rename a section using the Site Hierarchy pane. You can edit other values, like the ID, URL, and security using the Properties pane (see [Section 12.1.11, "Viewing and Editing Web Site and Section Properties"](#)).

To rename a section in the site hierarchy, perform these tasks:

1. In the Site Hierarchy pane, right-click the desired section and choose **Rename Section**.
2. Type a new name for the section, and then press ENTER on your keyboard or click anywhere in the Site Hierarchy pane.
3. To update the navigation of your site so that your changes take effect immediately, click the **Update Navigation** icon ([Figure 12-1](#)) on the Site toolbar. If you do not update the navigation, you are prompted to do so when you preview the site.

Figure 12-1 Update Navigation icon



12.1.3 Rearranging Sections in the Site Hierarchy

You can change the order in which sections appear in the site hierarchy, which affects their display order in the site navigation on your site. You do this by using a right-click, drag-and-drop mouse action in the Site Hierarchy pane in Designer.

To move a section to a new location in the site hierarchy, perform these tasks:

1. In the Site Hierarchy pane, right-click the desired section. Holding down the mouse, drag the section to the new location in the site hierarchy and let go of the mouse.
2. On the popup menu that appears, click **Move Before**, **Move After**, or **Move Into**, to move the section before, after, or into the section that was highlighted when you let go of the mouse button.
3. Click **OK** to the move warning message.

Using drag-and-drop with the left mouse button performs the same action as **Move Into** in the above steps.

The new order of sections opens in your web site navigation as soon as you update the navigation. You are prompted to update the navigation file when you preview the page template. You can also update this file manually by clicking the **Update Navigation** icon ([Figure 12-2](#)) on the Site toolbar.

Figure 12-2 Update Navigation icon



12.1.4 Removing a Section From the Site Hierarchy

You can remove sections from your web site using the Site Hierarchy pane in Designer.

To remove a section, perform these tasks:

1. In the Site Hierarchy pane, right-click the section to remove and click **Delete Section**.
2. Click **OK** to confirm the deletion.

If this section was also included in your site navigation, the link is removed from the web site when you update the navigation. You are prompted to update the navigation file when you preview the page template. You can also update this file manually by clicking the **Update Navigation** icon (Figure 12–3) on the Site toolbar.

Figure 12–3 Update Navigation icon



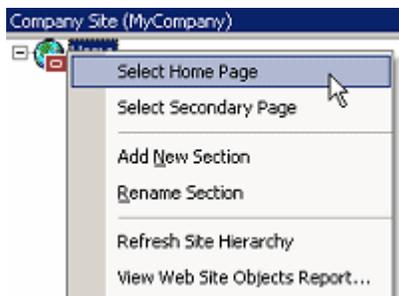
12.1.5 Assigning a Home Page

The home page of the web site is the entry. Everyone who comes to your web site usually starts at your home page.

To assign a page template to use as the home page for a section, perform these tasks:

1. In the Site Hierarchy pane, right-click the web site and choose **Select Home Page** from the popup menu.

Figure 12–4 Select Home Page option in popup menu



2. In the Select (Home) page template dialog (see [Section A.11, "Select \(Home\) Page Template Dialog"](#)), choose how you want to create the home page:
 - **Create a new page template:** A new page template is created, checked into the content server, and associated with this section. Choose **hcsp**, **jsp**, or **asp** as the file extension.

The available extension depends on the site type that you chose when creating the site.

- **Select existing page template from the content server:** An existing page template in the content server is associated with this section. Select the page template in the menu or select **<Choose From Server>** to browse to it in the content server when this dialog closes.
- **Create a new page template from an existing one:** A new HCSP, JSP, or ASP page is created based on another page template, checked into the content

server, and associated with the web site. The page template that is used to create the new one can come from one of two sources:

- **Stored in the content server:** A page template stored on the content server is used. Select it from the menu (if the page template is one you created recently); if not, select **<Choose From Server>** to browse to it in the content server when this dialog closes.
 - **Stored on the file system:** A page template stored on the file system is used. This option can be useful if you've created a page template on your file system (perhaps using a third-party editor) and want to use that design for this page template. The file must have the appropriate file extension (hcsp, jsp, or asp).
3. If you want to use this same page template in every section of the web site below this one, check the box **Apply recursively to all subsections**. (This performs the same action as if you're reusing a page template throughout the site hierarchy.)
 4. Click **OK** to close the Select (Home) Page Template dialog.

Depending on your choices above, you may be taken to the content server screen to select an existing page template or assign metadata to a new page template.

You can add a secondary page to the root of the site hierarchy as well and it serves as a backdrop for new files added by contributors.

Oracle Site Studio automatically assigns the value "Home" to the web site, which directs users to this page. If you want to change this value, select the web site in the Site Hierarchy pane, open the Properties pane, and enter a different value for **Label**.

A home page is different from a primary or secondary page in the following ways: First, it is associated with the root of your site instead of a section. Second, the home page link in some navigation fragments takes you to this page and no other page template. Third, the "Websites" menu in the content server takes you to this page of the web site.

You should pay close attention to the names and metadata values (*content ID, type, title, security group*, and so on) that you assign to your page templates. This helps you better manage your page templates, especially if you start reusing them in your site hierarchy.

12.1.6 Assigning a Primary Page

A primary page acts like a landing or default page for each section of the web site. When users click a link to go to that section (for example, Products), they are taken to the primary page associated with that section.

Both primary and secondary pages provide true scalability for a growing web site. You can manage the "look and feel" of your entire site, regardless of its size, with just a few page templates.

To assign a page template to use as the primary page for a section, perform these tasks:

1. In the Site Hierarchy pane, right-click the desired section and choose **Select Primary Page**.
2. In the Select (Primary) page template dialog (see [Section A.12, "Select \(Primary\) Page Template Dialog"](#)), choose how you want to create the primary page for this section:
 - **Create a new page template:** A new page template is created, checked into the content server, and associated with this section. Choose **hcsp, jsp, or asp** as the

file extension. The available extension depends on the site type that you chose when creating the site.

- **Select existing page template from the content server:** An existing page template in the content server is associated with this section. Select the page template in the menu or select **<Choose From Server>** to browse to it in the content server when this dialog closes.
 - **Create a new page template from an existing one:** A new HCSP, JSP, or ASP page is created based on another page template, checked into the content server, and associated with this section. The page template that is used to create the new one can come from one of two sources:
 - **Stored in the content server:** A page template stored on the content server is used. Select it from the menu (if the page template is one you created recently); if not, select **<Choose From Server>** to browse to it in the content server when this dialog closes.
 - **Stored on the file system:** A page template stored on the file system is used. This option can be useful if you've created a page template on your file system (perhaps using a third-party editor) and want to use that design for this page template. The file must have the appropriate file extension (hcsp, jsp, or asp).
 - **Use an external URL:** Uses the web page of another web site instead of a page template. As a result, visitors going to this section are directed to the web page URL that you specify here. (This option applies only to primary pages and is not available for a home page or a secondary page.)
3. Click **OK** to close the Select (Primary) Page Template dialog.

Depending on your choices above, you may be taken to the content server screen to select an existing page template or assign metadata to a new page template.

By default, the filename of the default URL page name is `index.htm`. By making a change to configuration flags in the `config.cfg` file, the administrator can select any value for the default URL page name. Additionally, other file names can be added to deliver the primary page. This is helpful if there are existing data files with path-based links using the previous default URL page name. For more information, see *Technical Reference for Oracle Site Studio*.

The only case when neither a primary page nor a secondary page is used by a section is if you choose to use an external URL in the Select Layout dialog. This option inserts the URL of another web site in the section instead of a page template. When visitors click a link for this section, they are taken to the web site that you specified. (This option is useful for linking to partner sites or department-level sites within the same organization.)

12.1.7 Assigning a Secondary Page

Each section in the site hierarchy, including the root of the site needs a secondary page if contributors are allowed to add new files (data files or native documents) to the web site. The secondary page acts like a backdrop for the new content added to the site by a contributor.

Both primary and secondary pages provide true scalability for a growing web site. You can manage the "look and feel" of your entire site, regardless of its size, with just a few page templates.

The secondary page in each section, if you decide to use one, is the template for the pages that can have dynamically placed content. A secondary page acts like a

backdrop for new files added to the site by a contributor. The page template is reused for each new data file or native document added to the site, and therefore you do not have to create multiple web pages in each section; just one primary page and one secondary page.

To assign a page template to use as the secondary page for a section, perform these tasks:

1. In the Site Hierarchy pane, right-click the desired section (or the root) and choose **Select Secondary Page**.
2. In the Select (Secondary) Page Template dialog (see [Section A.13, "Select \(Secondary\) Page Template Dialog"](#)), choose how you want to create the secondary page for this section:
 - **Create a new page template:** A new page template is created, checked into the content server, and associated with this section. Choose **hcsp**, **jsp**, or **asp** as the file extension. The available extension depends on the site type that you chose when creating the site.
 - **Select existing page template from the content server:** An existing page template in the content server is associated with this section. Select the page template in the menu or select **<Choose From Server>** to browse to it in the content server when this dialog closes.
 - **Create a new page template from an existing one:** A new HCSP, JSP, or ASP page is created based on another page template, checked into the content server, and associated with this section. The page template that is used to create the new one can come from one of two sources:
 - **Stored in the content server:** A page template stored on the content server is used. Select it from the menu (if the page template is one you created recently); if not, select **<Choose From Server>** to browse to it in the content server when this dialog closes.
 - **Stored on the file system:** A page template stored on the file system is used. This option can be useful if you've created a page template on your file system (perhaps using a third-party editor) and want to use that design for this page template. The file must have the appropriate file extension (hcsp, jsp, or asp).
3. If you want to use this same secondary page in every section of the web site below this one, check the box **Apply recursively to all subsections**. (This performs the same action as if you're reusing a page template throughout the site hierarchy.)
4. Click **OK** to close the Select (Secondary) Page Template dialog.

Depending on your choices above, you may be taken to the content server screen to select an existing page template or assign metadata to a new page template.

12.1.8 Showing Sections in the Site Navigation

Oracle Site Studio converts each section in the site hierarchy into a hyperlink on a navigation menu (created by a navigation fragment) on your site. This is done automatically when you include a section in the site navigation.

By default, new sections are not included in the site navigation and therefore are not automatically added as links to the web site. In the Site Hierarchy pane, the sections with the blank folder icon ([Figure 12-5](#)) are not included in the site navigation, and the sections with the colored folder icon ([Figure 12-6](#)) are included in the site navigation.

Figure 12–5 Blank Folders - Not Included in Site Navigation**Figure 12–6 Colored Folders - Included in Site Navigation**

To show a section in the site navigation, perform these tasks:

1. In the Site Hierarchy pane, right-click the desired section and click **Include Section in Navigation**.
2. Click the **Update Navigation** icon (Figure 12–7) on the Site toolbar to add a link to this section in the site navigation.

Figure 12–7 Update Navigation icon

You can also update the site navigation by clicking **Yes** when prompted to regenerate the navigation.js file when previewing a page template.

You can also specify whether a section is included in the site navigation by using the Properties pane (see [Section 12.1.11, "Viewing and Editing Web Site and Section Properties"](#)).

Some navigation fragments in the Toolbox were designed to show only second or third-level sections in the hierarchy (for example, see [Section C.1.8, "Nav Plain Vertical"](#)). If you use one of these fragments but do not have a hierarchy two or three levels deep, you may not see your sections in the navigation menu.

Reasons You May Not Want a Section in Your Site Navigation

There are a few reasons why you wouldn't include a section in your site navigation. Here are a few examples:

- **When adding a search feature** to your site, you must add a search box (see [Section C.1.11, "Search Box Plain"](#)) to your page templates, and you must designate a search results page template, which contains the search results (see [Section C.1.13, "Search Result Plain"](#)). The search results page should be in a section that is not included in your site navigation; otherwise, users would be able to click a link in your site navigation and go directly to the search results page, which would be empty.
- **When creating a custom error page**, you create a section in the site hierarchy that serves as a custom error handler (see [Section 12.1.9, "Designating a Section as an Error Handler"](#)). Then you create a page template in that section that opens the custom error message. This section should not be included in the site navigation because users would be able to click a link in your site navigation and go directly to the custom error page, which wouldn't make sense to the user.
- **When introducing new features or page templates**, you may choose to put them in a section that is not included in the site navigation so that others do not see your unfinished work. You can continue testing them until the features or page template and related site assets are ready. Then you can include that section in the site navigation.

As an alternative to not including a section in the site navigation, you can create a contributor-only section so that the section displays only in contribution mode.

12.1.9 Designating a Section as an Error Handler

There are several circumstances when a visitor or contributor to your site encounters an error message.

For example, if a secondary page template is missing in the section where a contributor data file is stored, you may encounter a message saying the secondary page could not be found. Or, if a contributor data file or native document is removed from the site but still being linked to from another part of the site, you may get a missing file message.

You can prevent such errors from appearing by designating a section as an error handler and then creating a custom error page (page template) for that section.

To designate a section as an error handler, perform these tasks:

1. Add a new section to your web site (see [Section 12.1.1, "Adding a Section to the Site Hierarchy"](#)).

Note: The "label" that you use does not appear anywhere on the site.

2. Right-click the section and choose **Set Section as Error Handler**.

The icon for the section displays in red with an exclamation mark ([Figure 12–8](#)).

Figure 12–8 Error Handler Section icon



3. Right-click the section again, choose **Select Primary Page**, and follow the steps to add a page template to the section (see [Section 12.1.6, "Assigning a Primary Page"](#)).
4. Then, on the page template, create the custom error message or use the sample fragment provided in the Toolbox (see [Section C.4.4, "Error Handler Message"](#)).

When you designate a section as an error handler, you do not want to include that section in your site navigation.

Using the Error Handler with Data Files

If the person viewing the page does not have security access to the data file or data files used on the page, you can select to either have the page display an error, or to simply not show the data. If the contributor does not have access, an error message can display in the space designated for the data file. This is controlled by controlling the data access check through the configuration flags in the *config.cfg* file. For more information, see *Technical Reference for Oracle Site Studio*.

12.1.10 Changing the Security of the Site or a Section

You can change the security for a particular section or the entire site, if you like. The security of the site uses the security model in the content server.

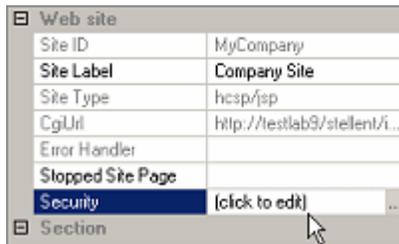
This can be useful when you want to specify which users in your organization have access to the site hierarchy and prevent unauthorized users from changing it.

To change the security for a section or your site, perform these tasks:

1. In the Site Hierarchy pane, select the root of the site or a section in the site.
2. Open the Properties pane.

- Click the **Security** property (Figure 12–9).

Figure 12–9 Security property

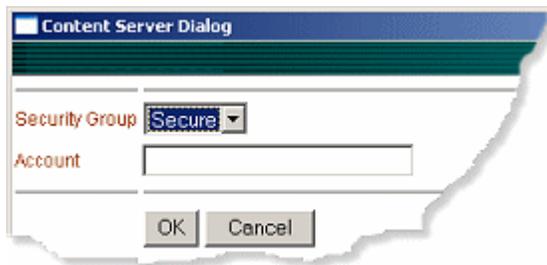


- Click the **Additional Information** icon (Figure 12–10) to open the Content Server dialog (Figure 12–11).

Figure 12–10 Additional Information icon



Figure 12–11 Content Server dialog



- In the **Security Group** menu, choose the group to assign to your site or section.
- In the **Account** field, specify an account for the site or section.
- Click **OK**.

12.1.11 Viewing and Editing Web Site and Section Properties

You can view all the properties and change certain properties for your site and each section using the Properties pane. This can be useful for managing and editing your site hierarchy.

To view the properties of the site or a section in the site, perform these tasks:

- In the Site Hierarchy pane, select the root of the site or the desired section in the site.
- Open the Properties pane to view the properties associated with the site and each section.

The properties that are grayed out (dimmed) cannot be changed in the Properties pane. All other properties can be modified directly. The values for each may vary.

Web Site

Property	Definition
Site ID	The ID of the site. This is the unique identifier for the site in the content server. The value also appears in the site address if you use a folder-based address to access the site.
Site Label	The name (and friendlier description) of the web site. The site label appears in Designer and the "Websites" menu in the content server.
Site Type	The scripting format for the site (HCSP/JSP or ASP).
CgiUrl	The Cgi URL address of the content server that hosts the site.
Error Handler	The name of the section serving as an error handler for the site (see Section 12.1.9, "Designating a Section as an Error Handler").
Stopped Site Page	The page that appears when the site is stopped (see Section 12.2, "Working With Fragments").
Security	Controls the security of the site (see Section 12.1.10, "Changing the Security of the Site or a Section").

Note: You can override the Site ID value by entering a value for "URL Directory Name" with the root of the site selected. This enables you to change the name that appears in a folder-based site address.

Section

Property	Definition
Section ID	The ID of the section (also referred to as a Node ID). This is used by Oracle Site Studio to call up a section of the site.
Label	The name that you chose for the section. It displays in the Site Hierarchy pane and your site navigation.
Include in Navigation	If set to True, the section displays in the site navigation. If set to False, the section does not display in the site navigation. For more information, see Section 12.1.8, "Showing Sections in the Site Navigation."
URL Page Name	The URL that represents the primary page in this section. If you don't specify a name, "index.htm" is used.
URL Directory Name	The URL that represents this section in the web site address .
Primary Page	The primary page associated with this section.
Primary Page Params	If the primary page contains a contribution region, the name of that region and the contributor data file assigned to it, display here.
Primary Page Region Templates	The template override specified for the primary page in this section.
Primary Page Placeholder Definition Mappings	The placeholder definition mappings for the primary page in this section of the web site. For more information on the placeholder mappings, see Section 9.8, "Working With Placeholder Definitions."
Secondary Page	The secondary page associated with this section.
Secondary Page Params	If the secondary page contains a contribution region, the name of that region and the contributor data file assigned to it, display here.

Property	Definition
Secondary Page Region Templates	The template override specified for the secondary pages in this section.
Secondary Page Placeholder Definition Mappings	The placeholder definition mappings for the secondary pages in this section of the web site. For more information on the placeholder mappings, see Section 9.8, "Working With Placeholder Definitions."
Replaceable Region	If the page template contains a replaceable region, the name of that region displays here.
Contributor-Only Section	If set to True , the section opens in the site navigation when the web page is viewed in contribution mode. When set to False , the section always appears on the site.
Ready to Replicate	If set to True , the section will be replicated. If set to False , the section will not be replicated. This section is present only if Enable Section Level Replication has been selected from the Advanced submenu. For more information on the Advanced submenu, see Section 5.9.1, "File Menu."
Max Age (caching)	Specifies a maximum amount of time that the primary page can be cached. This option is useful for setting up a reverse proxy cache for your site.
Max Age Secondary	Specifies a maximum amount of time that the secondary page can be cached. This option is useful for setting up a reverse proxy cache for your site.
Security	Controls the security of the section (see Section 12.1.10, "Changing the Security of the Site or a Section").

Note: The name used for secondary pages (the equivalent to "URL Page Name") is derived from the content ID (dDocName) of the data file or native document appearing on that page. For information on changing this value, see *Technical Reference for Oracle Site Studio*.

Custom Section Properties

If you've created one or more custom section properties in your site hierarchy, you see this third category of properties. When you create a custom section property, you can specify a unique value for each section in the site hierarchy and then refer to that property in your page templates and fragments (see [Section 12.1.12, "Defining Custom Properties for a Section"](#)).

12.1.12 Defining Custom Properties for a Section

Each section in the site hierarchy contains several properties, such as a label, the names of the page templates associated with it, security, and so on (see [Section 12.1.11, "Viewing and Editing Web Site and Section Properties"](#)).

In addition to these values, you can define your own custom property for each section in the site hierarchy and then modify your web site to read and execute those values. This becomes especially useful when you want to customize the appearance and

behavior of a particular section in the site hierarchy, such as changing the way a hyperlink for a section opens in a navigation fragment.

To define a custom section property, perform these tasks:

1. In the toolbar, click the **Custom Section Properties** icon (Figure 12–12) (or click **Tools** then select **Define Custom Section Properties**).

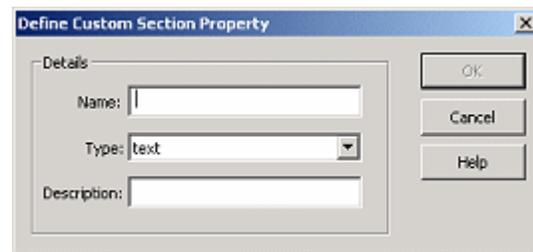
Figure 12–12 Custom Properties icon



2. In the Define Custom Section Properties dialog (see Section A.9, "Custom Section Properties Dialog"), click **Add**.

The Define Custom Section Property dialog opens (Figure 12–13). For more information, see Section A.10, "Define Custom Section Property Dialog."

Figure 12–13 Define Custom Section Property dialog



3. Enter a name in the **Name** field. The name displays in the Properties pane when you select the section in the site hierarchy.
4. Choose a value for **Type**. This determines the action to take (for a description of each type, see Section A.10, "Define Custom Section Property Dialog").
5. Enter a description in the **Description** field (optional). This displays in the description area at the bottom of the Properties pane.
6. Click **OK** twice to close both of the dialogs and return to Designer.

The custom property displays in the Properties pane for each section in the site hierarchy (see Section 12.1.11, "Viewing and Editing Web Site and Section Properties").

After you define a custom section property, you must go through the site hierarchy and enter a value for each section. Then, you must add the necessary code to your fragments or page templates to read and execute the custom property that you've added (see Section C.1.26, "CSP Sample Navigation (client)").

12.1.13 Expanding and Collapsing the Site Hierarchy

You can expand and collapse the site hierarchy in the Site Hierarchy pane to see all of the sections and page templates at once or to hide them from view. You use the toolbar in Designer to do this.

To expand and collapse the site hierarchy, perform these tasks:

- To expand the site hierarchy, click the **Expand Site Hierarchy** icon (Figure 12–14) in the toolbar.
- To collapse the site hierarchy, click the **Collapse Site Hierarchy** icon (Figure 12–15) in the toolbar.

Figure 12–14 Expand Site Hierarchy icon**Figure 12–15 Collapse Site Hierarchy icon**

12.1.14 Refreshing Your Site Hierarchy in Designer

As you work in Designer, there may be times when your site hierarchy becomes out of sync with the project file on the content server. To sync the two up and to make sure you're working with the most current project file, you can refresh the site hierarchy in Designer.

To refresh the site hierarchy in Designer, perform one of these tasks:

- Click the **Refresh** icon (Figure 12–16) in the Designer toolbar.

Figure 12–16 Refresh icon

- Right-click the root of the site in the Site Hierarchy pane and choose **Refresh Site Hierarchy**.

The icon in the lower right corner of Designer turns into the **Synchronized** icon (Figure 12–17) to indicate the site hierarchy is synchronized with the project file.

Figure 12–17 Synchronized icon

12.2 Working With Fragments

Fragments provide an additional tool for building and maintaining web sites. A fragment is like a packaged feature or piece of functionality that can be managed separately from the page template both before it is added and after it is added.

Fragments are an option for content that must be dynamic, requires frequent updates, and reused throughout a web site or multiple web sites. You can turn just about any piece of content, from a simple line of text to a server-side script, into a fragment.

Fragments are a powerful part of Oracle Site Studio. For more information, see [Chapter 13, "Working With Fragments."](#)

12.3 Assigning Content With The Switch Content Wizard

After the site has been constructed, what you have is a framework of the complete web site that is still missing the content. The contributor data files contain the content and are edited by the contributors. One way to add a contributor data file is to leave the task to the contributor to add the content. The other is for the designer to add the content. It is also possible to switch the current content file in a region for a different content file. Adding content, editing content, or switching contributor data files is done through the switch content wizard.

For the designer to add the content to a contributor data file, or to switch the contributor data file, follow these steps:

Open the Assign Content window, either by right-clicking and selecting **Assign Content** or highlighting a page template in the site hierarchy and clicking the **Assign Content** icon from the toolbar (see [Section A.54, "Assign Content Dialog"](#)).

The contribution graphic ([Figure 12–18](#)) shows a green plus sign if there is no content assigned.

Figure 12–18 Contribution graphic showing no content is assigned



Click the **Edit** icon with the plus sign, and the switch content wizard is launched (see [Section A.76, "Switch Content Wizard"](#)). If there is content already assigned, then instead of a green plus sign, there is a yellow arrow in the contribution graphic ([Figure 12–19](#)).

Figure 12–19 Contribution graphic showing switch region arrow



With the switch content wizard, it is simple to create and assign new content, or to assign content that exists on the server. You can also assign content that is stored locally.

To assign content, follow these steps:

1. Right-click a page template in the site hierarchy to open the Assign content window (see [Section A.54, "Assign Content Dialog"](#)).

The page opens in the dialog showing the contribution graphics.

2. Click the contribution graphic showing the green plus, or the yellow arrow, depending on whether content has been assigned.

The Switch Content wizard opens (see [Section A.76, "Switch Content Wizard"](#)).

3. The options available depend on the region definitions and subtemplates listed in the placeholder definition.

If you select a subtemplate, then the wizard finishes. If there are placeholders within the subtemplate then you must run the wizard for each placeholder.

If you select a region definition, then the wizard continues.

4. On the next screen of the wizard, you can add a new contributor data file, a new native document, an existing file on the content server, or a file on the local computer.

If you select any file not existing on the server, then you are asked to check the file into the server before the content is associated.

If you select a file on the server, a window opens with search results from the content server. Select a file and it is associated.

5. After the wizard is completed, the page reloads within the assign content window, additionally displaying the data associated through the wizard.

12.4 Working With Links

Links provide a convenient way for visitors to navigate from one web page to another. The most common use of a link is in the site navigation, such as a top menu, side menu, breadcrumbs, and so on. In Oracle Site Studio, you can instantly create these menus using a navigation fragment (see [Section C.1, "Navigation Fragments"](#)).

There are, however, instances when you may want to create additional links to take users to a specific piece of content. You can add a link to another section of your site, to a contributor data file or native document on your site, to another web site, and more. You do this using the Link wizard.

In addition to creating links in Designer, you can allow contributors to create and edit links in Contributor. You set this up when you add an element to a contribution region.

The Link wizard takes you through all variations on linking:

- [Section 12.4.1, "Creating a Link to Another Section of the Web Site"](#)
- [Section 12.4.2, "Creating a Link to an Existing File on the Content Server"](#)
- [Section 12.4.3, "Creating a Link to a URL"](#)
- [Section 12.4.4, "Creating a Link to a New Content File"](#)
- [Section 12.4.5, "Creating a Link to an Existing Local File"](#)
- [Section 12.4.6, "Editing a Hyperlink"](#)
- [Section 12.4.7, "Removing a Link"](#)
- [Section 12.4.8, "Choosing a Default Link Format"](#)
- [Section 12.4.9, "Hyperlink Formats"](#)

12.4.1 Creating a Link to Another Section of the Web Site

To create a link to another section of your web site or another Oracle Site Studio site on the content server, perform these tasks:

1. Highlight the text (or image) that you would like to turn into a link.
2. Click the **Create Hyperlink** icon ([Figure 12-20](#)) in the toolbar.

Figure 12-20 Create Hyperlink Icon



The Link wizard is launched (see [Section A.18, "Link Wizard"](#)).

3. Select **Link to a section**.
4. Click **Next**.

Note: You can step through the wizard faster by clicking the blue arrow to the left of the option you want to use. This automatically selects the option and moves the wizard to the next step.

5. Choose the section that you would like to link to. This can be a section on the current web site or a different Oracle Site Studio site on the content server. You select the target section in the site hierarchy.

6. Click **Next**.
7. Depending on the site configuration, you may be prompted to choose the URL format of the link:
 - **Path Based URL:** The link will contain a path to the target location. You have two choices for this type of link:
 - **Absolute Path:** Generates a full path; for example
`<!--$ssServerRelativeSiteRoot-->products/index.htm` (where "`<!--$ssServerRelativeSiteRoot-->`" gets replaced with the path to the root of the web site).
 - **Relative Path:** Generates a relative path instead of a full path; for example
`../products/index.htm`.

Please note that this option is not available if you chose to use the default web site section for the content file as the target section (see previous step).
 - **ID Based URL:** The link will contain the coded identity of the target location rather than the path-based name. You have three choices for this type of link:
 - **Client Side Script Format:** Uses client-side JavaScript to construct a link to the target location; for example, `javascript:nodelink('10024');`
 - **Server Side Script Format:** Uses server-side Idoc Script to construct a link to the target location; for example,
`[!--$wcmUrl('nodelink','10024')--]`.
 - **URL Token Format:** Uses a redirect on the server to construct a link to the target location; for example, `ssNODELINK/10024`.
8. Click **Options...**
 The Link Target Options dialog opens.
9. If you want the link to open in a new web browser window when clicked, select the **Open link target in new window** checkbox, and then choose the target in the menu (`_blank`). You can also directly enter the name of the target in the text box if you want, but if it is not a valid target, the link is faulty.
10. Click **OK** or **Cancel** to return to the wizard.
11. Click **Next**.
12. Verify the URL for the link.
13. Click **Finish**.

12.4.2 Creating a Link to an Existing File on the Content Server

You can create a link to an existing content file (contributor data file or native document) on your site or another site in the content server.

To create a link to an existing file on the content server, perform these tasks:

1. Highlight the text (or image) that you would like to turn into a link.
2. Click the **Create Hyperlink** icon (Figure 12–21) in the toolbar.

Figure 12–21 Create Hyperlink Icon



The Link wizard is launched (see [Section A.18, "Link Wizard"](#)).

3. Select **Link to a file**.
4. Click **Next**.

Note: You can step through the wizard faster by clicking the blue arrow to the left of the option you want to use. This automatically selects the option and moves the wizard to the next step.

5. Select **Existing file from server**, and click **Next**.

The standard search results page on the content server opens, showing all content files associated with the web site.

Note: The content server query from the link wizard is an open query in Designer. In Contributor, the query is more controlled.

6. Click **Select** next to the file you want to link to.
7. Choose the target section for the link. You can choose from the following options:
 - **Use default website section metadata:** If you choose this option, the hyperlinked content file will display in the web site section as currently specified in the Web Site Section metadata field for the content file.
 - **Choose a website section:** If you choose this option, the hyperlinked content file will display in the web site section of your choice (in effect overriding the default target section of the content file). The section may be in the current web site or another Oracle Site Studio site on the content server.
 - **Link to the Content Item's URL:** If you choose this option, the file will display exactly as it is stored on the content server rather than within the framework of the web site. This is useful for linking to native documents such as PDF files.

Click **Next** to continue.

8. Depending on the site configuration, you may be prompted to choose the URL format of the link:
 - **Path Based URL:** The link will contain a path to the target location. You have two choices for this type of link:
 - **Absolute Path:** Generates a full path; for example
`<!--$ssServerRelativeSiteRoot-->Menus/switched_region_CDF` (where "`<!--$ssServerRelativeSiteRoot-->`" gets replaced with the path to the root of the web site).
 - **Relative Path:** Generates a relative path instead of a full path; for example
`../Menus/switched_region_CDF`.
 - Please note that this option is not available if you chose to use the default web site section for the content file as the target section (see previous step).
 - **ID Based URL:** The link will contain the coded identity of the target location rather than the path-based name. You have three choices for this type of link:

- **Client Side Script Format:** Uses client-side JavaScript to construct a link to the target location; for example, `javascript:link('switched_region_CDF');`.
- **Server Side Script Format:** Uses server-side Idoc Script to construct a link to the target location; for example, `<!--$wcmUrl("link","switched_region_CDF")-->`.
- **URL Token Format:** Uses a redirect on the server to construct a link to the target location; for example, `ssLINK/switched_region_CDF`.

9. Click **Options...**

The Link Target Options dialog opens.

10. If you want the link to open in a new web browser window when clicked, select the **Open link target in new window** checkbox, and then choose the target in the menu (`_blank`). You can also directly enter the name of the target in the text box if you want, but if it is not a valid target, the link is faulty.
11. Click **OK** or **Cancel** to return to the wizard.
12. Click **Next**.
13. Verify the URL for the link.
14. Click **Finish**.

12.4.3 Creating a Link to a URL

You can create a link to a specific URL (web address); for example, an Internet web site or a file on the content server (one not used for contribution, such as a PDF file, media file, or zip file).

To create a link to a specific URL, perform these tasks:

1. Highlight the text (or image) that you would like to turn into a link.
2. Click the **Create Hyperlink** icon (Figure 12–22) in the toolbar.

Figure 12–22 Create Hyperlink Icon



The Link wizard is launched (see Section A.18, "Link Wizard").

3. Select **Link to following URL**.
 4. Enter the web address in the field provided (for example, `http://www.oracle.com`).
- You cannot proceed in the wizard until you provide a URL.

5. Click **Options...**

The Link Target Options dialog opens.

6. If you want the link to open in a new web browser window when clicked, select the **Open link target in new window** checkbox, and then choose the target in the menu (`_blank`). You can also directly enter the name of the target in the text box if you want, but if it is not a valid target, the link is faulty.

Note: When creating a link to another web site, the "Open link target in new window" option may be useful because it prevents users from inadvertently leaving your web site when they click the link.

7. Click **OK** or **Cancel** to return to the wizard.
8. Click **Next**.
9. Verify the URL for the link.
10. Click **Finish**.

12.4.4 Creating a Link to a New Content File

You can create a link to a new content file (contributor data file or native document), which is checked into the content server as a managed site asset.

To create a link to a new content file, perform these tasks:

1. Highlight the text (or image) that you would like to turn into a link.
2. Click the **Create Hyperlink** icon (Figure 12–23) in the toolbar.

Figure 12–23 Create Hyperlink Icon



The Link wizard is launched (see [Section A.18, "Link Wizard"](#)).

3. Select **Link to a file**.
4. Click **Next**.

Note: You can step through the wizard faster by clicking the blue arrow to the left of the option you want to use. This automatically selects the option and moves the wizard to the next step.

5. Select either **New Contributor data file** or **New native file**, depending on which type of file you want to add. For a new native document, also select the file format from the menu. The site designer decides what file formats are allowed for new native documents.

6. Click **Next**.

The standard content check-in page on the content server opens.

7. Enter the appropriate content information (metadata) to check the item into the content server. Much of the information is automatically provided. If a field name is marked red, then the field is required (that is, the item cannot be checked in if the field is empty). Please note that the new contributor data file or native document is empty until it is edited, usually by a contributor.

Click **Next** when you are done.

8. Choose the target section for the link. You can choose from the following options:
 - **Use default website section metadata:** If you choose this option, the hyperlinked content file will display in the web site section as currently specified in the Web Site Section metadata field for the content file.

- **Choose a website section:** If you choose this option, the hyperlinked content file will display in the web site section of your choice (in effect overriding the default target section of the content file). The section may be in the current web site or another Oracle Site Studio site on the content server.
- **Link to the Content Item's URL:** If you choose this option, the file will display exactly as it is stored on the content server rather than within the framework of the web site. This is useful for linking to native documents such as PDF files.

Click **Next** to continue.

9. Depending on the site configuration, you may be prompted to choose the URL format of the link:

- **Path Based URL:** The link will contain a path to the target location. You have two choices for this type of link:
 - **Absolute Path:** Generates a full path; for example
`<!--$ssServerRelativeSiteRoot-->Menus/switched_region_CDF` (where "`<!--$ssServerRelativeSiteRoot-->`" gets replaced with the path to the root of the web site).
 - **Relative Path:** Generates a relative path instead of a full path; for example
`../Menus/switched_region_CDF`.

Please note that this option is not available if you chose to use the default web site section for the content file as the target section (see previous step).

- **ID Based URL:** The link will contain the coded identity of the target location rather than the path-based name. You have three choices for this type of link:
 - **Client Side Script Format:** Uses client-side JavaScript to construct a link to the target location; for example, `javascript:link('switched_region_CDF');`
 - **Server Side Script Format:** Uses server-side Idoc Script to construct a link to the target location; for example, `<!--$wcmUrl("link","switched_region_CDF")-->`.
 - **URL Token Format:** Uses a redirect on the server to construct a link to the target location; for example, `ssLINK/switched_region_CDF`.

10. Click **Options...**

The Link Target Options dialog opens.

11. If you want the link to open in a new web browser window when clicked, select the **Open link target in new window** checkbox, and then choose the target in the menu (`_blank`). You can also directly enter the name of the target in the text box if you want, but if it is not a valid target, the link is faulty.
12. Click **OK** or **Cancel** to return to the wizard.
13. Click **Next**.
14. Verify the URL for the link.
15. Click **Finish**.

12.4.5 Creating a Link to an Existing Local File

You can create a link to an existing local file (that is, on your own computer), which is checked into the content server as a managed site asset.

To create a link to an existing local file, perform these tasks:

1. Highlight the text (or image) that you would like to turn into a link.
2. Click the **Create Hyperlink** icon (Figure 12–24) in the toolbar.

Figure 12–24 Create Hyperlink Icon



The Link wizard is launched (see Section A.18, "Link Wizard").

3. Select **Link to a file**.
4. Click **Next**.

Note: You can step through the wizard faster by clicking the blue arrow to the left of the option you want to use. This automatically selects the option and moves the wizard to the next step.

5. Select **Existing local file**, and click **Next**.

The standard content check-in page on the content server opens.

6. Enter the appropriate content information (metadata) to check the item into the content server. Much of the information is automatically provided. If a field name is marked red, then the field is required (that is, the item cannot be checked in if the field is empty). Use the **Browse** button next to the **Primary File** field to navigate to the file on your computer and select it. Also, make sure to specify a title.

Click **Next** when you are done.

7. Choose the target section for the link. You can choose from the following options:
 - **Use default website section metadata:** If you choose this option, the hyperlinked content file will display in the web site section as currently specified in the Web Site Section metadata field for the content file.
 - **Choose a website section:** If you choose this option, the hyperlinked content file will display in the web site section of your choice (in effect overriding the default target section of the content file). The section may be in the current web site or another Oracle Site Studio site on the content server.
 - **Link to the Content Item's URL:** If you choose this option, the file will display exactly as it is stored on the content server rather than within the framework of the web site. This is useful for linking to native documents such as PDF files.

Click **Next** to continue.

8. Depending on the site configuration, you may be prompted to choose the URL format of the link:
 - **Path Based URL:** The link will contain a path to the target location. You have two choices for this type of link:
 - **Absolute Path:** Generates a full path; for example `<!--$ssServerRelativeSiteRoot-->Menus/switched_region_CDF` (where `<!--$ssServerRelativeSiteRoot-->` gets replaced with the path to the root of the web site).

- **Relative Path:** Generates a relative path instead of a full path; for example `../Menus/switched_region_CDF`.

Please note that this option is not available if you chose to use the default web site section for the content file as the target section (see previous step).

- **ID Based URL:** The link will contain the coded identity of the target location rather than the path-based name. You have three choices for this type of link:
 - **Client Side Script Format:** Uses client-side JavaScript to construct a link to the target location; for example, `javascript:link('switched_region_CDF');`
 - **Server Side Script Format:** Uses server-side Idoc Script to construct a link to the target location; for example, `<!--$wcmUrl("link", "switched_region_CDF")-->`.
 - **URL Token Format:** Uses a redirect on the server to construct a link to the target location; for example, `ssLINK/switched_region_CDF`.

9. Click Options....

The Link Target Options dialog opens.

10. If you want the link to open in a new web browser window when clicked, select the **Open link target in new window** checkbox, and then choose the target in the menu (`_blank`). You can also directly enter the name of the target in the text box if you want, but if it is not a valid target, the link is faulty.
11. Click **OK** or **Cancel** to return to the wizard.
12. Click **Next**.
13. Verify the URL for the link.
14. Click **Finish**.

12.4.6 Editing a Hyperlink

After you add a hyperlink, you can edit the link by highlighting it and opening the Link Wizard.

To edit a hyperlink, perform these tasks:

1. In the page template, place your cursor on the text or graphic that is serving as a hyperlink.
2. Click the **Create Hyperlink** icon (Figure 12–25) in the Formatting toolbar.

Figure 12–25 Create Hyperlink Icon



3. This opens the Link Wizard, where you can go through and change the hyperlink in the same way that you added it.
4. Click **Finish** to close the wizard and return to the page template.

You can also edit a hyperlink by placing your cursor on the text or graphic that is serving as the link and editing the `href` attribute in the Properties pane.

12.4.7 Removing a Link

After you add a link using the Link wizard, you can easily remove it at any time, by performing these tasks:

1. Place your mouse cursor anywhere in the hyperlinked text (or highlight the text).
2. Click the **Remove Hyperlink** icon (Figure 12–26) in the toolbar.

The hyperlink is removed from the text.

Figure 12–26 Remove Hyperlink Icon



12.4.8 Choosing a Default Link Format

When you create a hyperlink using the Link wizard, you have numerous choices regarding the format of the link. The format is the way the link is assembled. The link can use a path-based URL or an ID-based URL to locate the target location.

Within each format, there are even more choices (for example, relative paths, client-side IDs, server-side IDs, and so on). There are benefits and drawbacks to each one (see Section 12.4.9, "Hyperlink Formats"). As such, you may want to choose a default format and then hide this option in the Link wizard; for both designers and contributors.

To choose a default link format, perform these tasks:

1. Click **File** then select **Site**, then select **Advanced** and then **Set Default Link Format**.
This opens the Choose Default Link Format dialog (see Section A.6, "Choose Default Link Format Dialog").
2. Choose **Path-based URL** if you want the link to use a path to the target location:
 - **Absolute path:** Generates a full path; for example `<!--$ssServerRelativeSiteRoot-->products/index.htm` (where `"<!--$ssServerRelativeSiteRoot-->"` gets replaced with the path to the root of the web site).
 - **Relative path:** Check this to generate a relative path instead of a full path; for example `../products/index.htm`.
3. Choose **ID-based URL** if you want the link to use the hard-coded ID of the target location without referencing a section name or label:
 - Choose **Client Side javascript:link() / javascript:nodelink() format** if you want the link to use client-side JavaScript and the ID of the target location.
 - Choose **Server Side wcmUrl() format** if you want the link to use server-side Idoc Script and the ID of the target location.
 - Choose **URL Token ssLINK / ssNODELINK format** if you want the link to use an Oracle Site Studio token and the ID of the target location.
4. Check **Hide the "Choose Format" page in the Link Wizard** to prevent this option from appearing when you create links in Designer.
5. Check **Hide the "Choose Format" page in the Link Wizard** to prevent this option from appearing when you create links in Contributor.
6. Click **OK**.

You might hide the "Choose Format" screen in the Link wizard for several reasons (for example, to prevent user confusion, to prevent undesired linking choices by the user, or to enforce consistency for the hyperlinks on your site).

12.4.9 Hyperlink Formats

When you create a hyperlink, you have numerous choices regarding the format of the link. The format is the way the link is assembled. You can choose a path-based URL for the link or an ID-based URL for the link.

If you choose path-based, then you can choose between a full path or a relative path. If you choose ID-based, then you can choose between client-side script, server-side script, or a special token provided by Oracle Site Studio.

There are advantages and disadvantages to each. Please review the following table for a side-by-side comparison.

Side-By-Side Comparison

Advantages	Full Path	Rel. Path	Server-Side ID	Client-Side ID	URL Token
Natural site address	Yes	Yes	Yes	No	No
Hides cryptic siteID and nodeID values	Yes	Yes	Yes	No	No
Can be indexed by search engines	Yes	Yes	Yes	No	Yes
Unaffected by site hierarchy changes	No	No	Yes	Yes	Yes
Works in reused page templates	Yes	No	Yes	Yes	Yes
Requires no server-side scripting	Yes	Yes	No	Yes	Yes
Requires no client-side scripting	Yes	Yes	Yes	No	Yes
Requires no redirects on server	Yes	Yes	Yes	No	No
Capable of bookmark links	Yes	Yes	Yes	No	No
Supports passing parameters	Yes	Yes	No	No	Yes
Can open in popup windows	Yes	Yes	Yes	No	Yes
Can be used in native documents	Yes	Yes	No	Yes	Yes

Description of Each Category

Category	Description
Natural site address	You see a friendly, path-based address in the URL.
Hides cryptic siteID and nodeID values	Visitors and contributors viewing the site do not see the cryptic Oracle Site Studio technology in the URL.

Category	Description
Can be indexed by search engines	Search engines can index the site.
Unaffected by site hierarchy changes	You can change the site hierarchy (for example, rename sections or move a section into another one) without affecting your links.
Works in reused page templates	You can reuse the page template, data file, or native document throughout the site, and the link still works.
Requires no server-side scripting	The web site does not have to rely on server-side script (for example, Idoc Script).
Requires no client-side scripting	The web site does not have to rely on client-side JavaScript.
Requires no redirects on server	The web site does not have to rely on a server-side redirect that points the ID-based address to a path-based address, which may result in decreased site performance.
Capable of bookmark links	You can use the link for bookmark links, which allow the user to link to any part of a web page.
Supports passing parameters	You can append, or pass, a parameter to the URL to achieve a desired result.
Can open in a popup window	You can use the link to open a new popup window, which requires some scripting.
Can be used in native documents	You can use this same format of a link in a native document.

12.5 Comparing Changes on a Web Page

As you edit page templates in the site hierarchy, there may be times when you want to see what has changed on a given page, before and after you (or someone else) edited that page. You may want to know, for example, how much text was removed, how much new text was added, and where most of the edits took place.

You can do this using the document comparison feature in Oracle Site Studio. You can use this feature to compare page templates in Designer, and you can use it to compare web pages from the contribution graphic on a web page. Depending on which one you use, you are comparing different results.

- **When comparing page templates in Designer**, you compare two versions of a page template: the one stored in memory and the latest one checked into the content server. In other words, you're seeing the changes you've made to the page before you save it.
- **When comparing web pages using the contribution graphic**, you compare two versions of a complete web page, which may comprise data files, native documents, page templates, fragments, and more. When you compare web pages like this, you compare the most current version of the page and the latest released version in the content server.

You only see a difference between the current version and the latest released version of a web page when the release date has been set for a future time, or the page is part of a workflow and hasn't been approved yet. For more information, see [Chapter 15, "Using Workflows."](#)

The following sections describe this feature in more detail:

- [Section 12.5.1, "Color-Coded Differences when Comparing Web Pages"](#)
- [Section 12.5.2, "Using the Compare Changes Feature"](#)

12.5.1 Color-Coded Differences when Comparing Web Pages

When you compare two versions of a page template in Designer or a web page on the web site, you need some kind of color-coding convention to see the actual differences. Oracle Site Studio uses the following convention to indicate content that has changed.

- New text appears in green highlight.
- Deleted text appears in red with a strikethrough.
- Text that has not changed remains in black.

Figure 12–27 Example of Page Template Edited in Designer

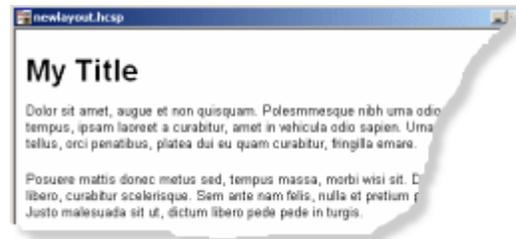
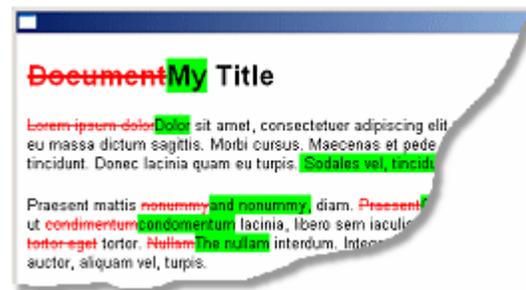


Figure 12–28 Same Page Template With Changes Highlighted



12.5.2 Using the Compare Changes Feature

You can compare the changes made to page templates in Designer, and you can compare the changes made to a web page on the web site (see [Section 12.5, "Comparing Changes on a Web Page"](#)).

Comparing Changes to Page Templates

To compare changes made to a page template in Designer, perform these tasks:

1. After making changes to a page template, click the **Compare Changes** icon ([Figure 12–29](#)) in the Site toolbar.

Figure 12–29 Compare Changes Icon



2. A new window opens showing the differences between the two versions of the page template (see [Section 12.5.1, "Color-Coded Differences when Comparing Web Pages"](#)).

This compares the unsaved version of the page template with the saved version in the content server. Take note of the changes and then close the window.

Comparing Changes to Web Page

To compare the changes made to a web page on the web site, perform these tasks:

1. Browse to the desired web page and open Contributor.
2. To compare the two versions of the web page (the most current version and the latest released version), click the **Menu** icon on the top of the page in contribution mode (Figure 12–30) and choose **View Differences**.

Figure 12–30 Contribution Mode Top Bar



3. A new window opens showing the differences between the two versions of the web page (see Section 12.5.1, "Color-Coded Differences when Comparing Web Pages").

You see only the changes in the new window if there is a difference between the most current version of the web page and the latest released version in the content server. Usually, this only happens when the page has a future release date or is part of a workflow (see Chapter 15, "Using Workflows"). Take note of the changes and then close the window.

When you compare two versions of a web page, you are comparing everything on the web page, which may include data files, native documents, page templates, fragments, and more.

This feature compares content changes, not formatting changes. For example, if you change the font size of the text or make the text bold, there are no differences displayed. But if you replace one word with another word, this difference is displayed. If an image is resized or replaced, the difference is displayed.

Working With Fragments

In Oracle Site Studio, a fragment can be a single line of text (such as a copyright notice), or it can be a complex collection of scripts performing client-side and server-side actions (such as a JavaScript-based menu).

This section covers the following topics:

- [Section 13.1, "About Fragments"](#)
- [Section 13.2, "Adding and Editing Fragments"](#)
- [Section 13.3, "Fragment Handling in Templates"](#)
- [Section 13.4, "Fragments in the Toolbox"](#)
- [Section 13.5, "Filtering Fragments in the Toolbox"](#)
- [Section 13.6, "Adding Fragments to a Template"](#)
- [Section 13.7, "Previewing Fragments"](#)
- [Section 13.8, "Editing Fragments"](#)
- [Section 13.9, "Creating a New Fragment"](#)
- [Section 13.10, "Deleting a Fragment"](#)
- [Section 13.11, "Deleting a Fragment in the Toolbox"](#)
- [Section 13.12, "Using the Fragment Editor"](#)
- [Section 13.13, "Fragment Libraries"](#)

13.1 About Fragments

When you first design a page template, you generally start with the basic look and feel of the site, using background colors, graphics, text, and a table layout. This kind of information opens in the same way on every page of the web site, or a section of the web site. You can, therefore, add this information directly to the page template, subtemplate, or region template in source or design view.

A fragment can be a single line of text (such as a copyright notice), or it can be a complex collection of scripts performing client-side and server-side actions (such as a JavaScript-based menu).

Just about anything that you add to a page template, subtemplate, or region template can be added as a fragment. How often you use fragments depends on your preference and the requirements of your web site.

13.2 Adding and Editing Fragments

Fragments can be added and edited using the Toolbox in Designer. Oracle Site Studio includes numerous sample fragments in the Toolbox that you can use or customize for your own (see [Section 13.4, "Fragments in the Toolbox"](#)). To use a fragment, simply place your cursor on the template where you would like the fragment to appear and then click the desired fragment in the Toolbox.

You can customize the way a fragment appears on a template by changing its parameters in the Fragment Parameter Values dialog, which you see when you add the fragment to a template.

The fragments contain parameters so that you can customize the fragment each time you use it on a template. If there's a setting that you're unable to change in a parameter, you can always open the fragment in the Fragment Editor and customize it there; even add your own custom parameter if you like.

13.3 Fragment Handling in Templates

Page templates, being complete HTML documents, have a <HEAD> and <BODY> in their code. Subtemplates and region templates, however, do not have a <HEAD> section. You must be aware of this distinction when placing a fragment, and also when creating and editing fragments.

The snippets inside of a fragment have a designated location of head, drop-point, top-of-body, or bottom-of-body. These designations are made when creating or editing fragments in the Fragment Editor dialog (see [Section A.57, "Fragment Editor Dialog"](#)). Those snippets designated with a head location must be placed on a template with a <HEAD>; that is, they must be placed on a page template because they require the header to work. If you place a fragment that has a snippet with a head location on a region template or a subtemplate, then the fragment cannot work.

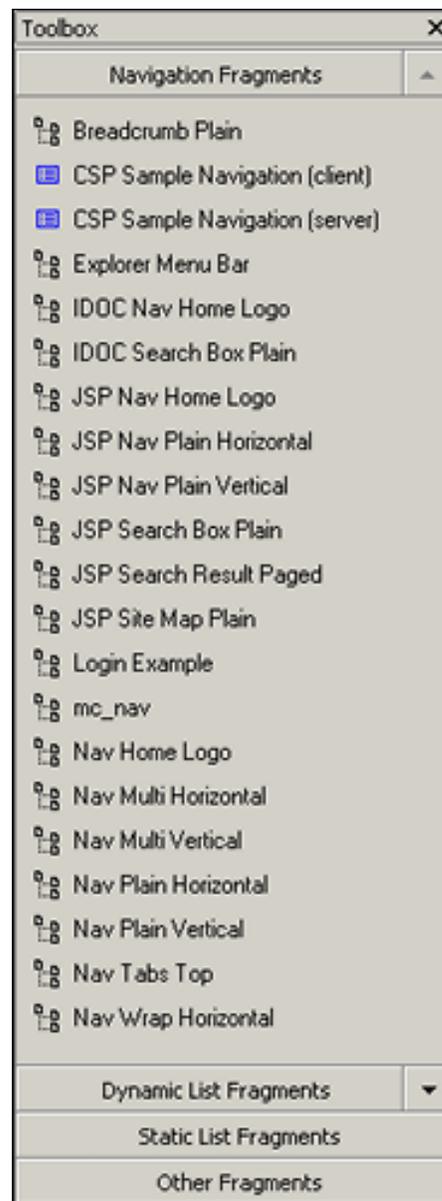
When you put a fragment on a page template, this includes information in the XML data island, specifically in the <SSINFO> tag. When you put a fragment on a subtemplate or a region template, then a wcmFragment tag is placed in the HTML. In both cases this can be seen in source view.

It is recommended that, whenever possible, you create fragments with a single body snippet. The fragment can then be used in any situation, whether it is on a page template, a subtemplate, or a region template.

13.4 Fragments in the Toolbox

Oracle Site Studio includes numerous out-of-the-box fragments that you can begin using right away without knowing how to create or edit a fragment. Using the sample fragments is a great way to add instant functionality to your web site while learning about the practical use and potential of a fragment.

Figure 13-1 Sample Fragments in the Toolbox



The sample fragments are available in the Toolbox and are divided into four categories:

- **Navigation Fragments:** This category contains fragments that provide site navigation for your web site (navigation bar, breadcrumbs, search box, and so on). The beauty of a navigation fragment is that it automatically reads your existing site hierarchy and converts the sections into hyperlinks that display in a navigation menu. Consequently, you can update and change your site hierarchy, and those changes are seen immediately on your web site. For a description of these fragments, see [Section C.1, "Navigation Fragments."](#)
- **Dynamic List Fragments:** This category contains fragments that provide advanced contribution features for your web site, specifically the ability to query for contributor data files and native documents in the content server. Contributors can then use the Contributor application to add, remove, and edit the files in the list (see [Section C.2, "Dynamic List Fragments"](#)).

- **Static List Fragments:** This category contains fragments that provide advanced contribution features for your web site, specifically the ability to create a structured layout of elements (WYISIWYG, plain text, image, and custom). Contributors can then use the Contributor application to add, remove, and edit rows and columns of information in the list (see [Section C.3, "Static List Fragments"](#)).
- **Other Fragments:** This category contains miscellaneous fragments that you may find useful for your site (a copyright line, an embedded Flash presentation, a dynamically converted native document, Manager, and so on). For a description of these fragments, see [Section C.4, "Other Fragments."](#)

13.5 Filtering Fragments in the Toolbox

The Toolbox in Designer displays all of the fragments that are available to you as you build your web site. As you and other designers introduce fragments to the site (or sites), however, the total number of fragments in the Toolbox can quickly become unmanageable.

You can control which fragments display in the Toolbox by filtering them based on their scripting language, whether they belong to your site or not, and whether you've introduced them or they came with Oracle Site Studio.

To filter fragments in the Toolbox, perform these tasks:

1. From the **View** menu, select **Customize**.
2. Click the **Miscellaneous** tab.
3. To display only the fragments written in the same language as your site (HCSP/JSP or ASP), check **Filter fragments in other languages**.
4. To display only the fragments that belong to your site (and not other sites), check **Filter libraries from other sites**.

Fragments in fragment libraries not identified as part of your site no longer appear in the Toolbox. To make a fragment part of your site, add it as a site asset to the "Fragment Libraries" category.

5. To display only the fragments that you created (and not the default ones that ship with Oracle Site Studio), check **Filter default libraries**.
6. Click **Close** to close the Customize dialog.

One reason you may want to show all of the available fragments on the server is to copy one of those fragment and make it part of your site.

13.6 Adding Fragments to a Template

You can use the Toolbox to add fragments directly to a template while working in design or source view.

To add a fragment, perform these tasks:

1. Place your cursor in the template where you want the fragment to appear.
If you're adding a list fragment, it may be necessary to first create a contribution region and then place the fragment in that region.
2. In the Toolbox, click a fragment category: **Navigation Fragments**, **Dynamic List Fragments**, **Static List Fragments**, or **Other Fragments**.

3. Click the desired fragment (for a description of each fragment, see [Appendix C, "Sample Fragments"](#)).
4. If the fragment contains parameters, you are prompted for additional information in the Fragment Parameter Values dialog (see [Section A.65, "Fragment Parameter Values Dialog"](#)).
5. Enter the appropriate values in the fields provided and then click **OK** to add the fragment.

The fragment displays on your template where your cursor was last located.

Many of the sample fragments insert a small text or a graphic as a visual aid so that you know where a fragment is located on a template. This small marker is known as the design view code. The design view code is used because many of the fragments contain dynamic content and server-side scripts that must be viewed in a web browser on the web site to be seen properly.

The placeholder does not necessarily represent how the fragment appears on your web site. You can change the placeholder in the Fragment Editor, if you like (see Step 5 in [Section 13.12.4, "Adding, Editing, and Deleting Fragment Snippets"](#)).

13.7 Previewing Fragments

Previewing a fragment can be useful to determine what the fragment looks like and how it works before making it a part of your template. You can preview fragments using the Preview option in the right-click menu of the Toolbox, or using the Preview button in the Fragment Editor (see [Section 13.12, "Using the Fragment Editor"](#)).

This section covers the following topics:

- [Section 13.7.1, "Previewing a Fragment"](#)
- [Section 13.7.2, "Choosing Different Parameters for a Fragment"](#)
- [Section 13.7.3, "Choosing Different Parameters When Adding a Fragment"](#)
- [Section 13.7.4, "Changing the Parameters of a Fragment"](#)

13.7.1 Previewing a Fragment

To preview a fragment, perform these tasks:

1. In the Toolbox, click a fragment category: **Navigation Fragments, Dynamic List Fragments, Static List Fragments, or Other Fragments**.
2. Right-click the desired fragment and choose **Preview**.
3. If the fragment contains parameters, you are taken to the Fragment Parameter Values dialog (see [Section A.65, "Fragment Parameter Values Dialog"](#)).

Enter the appropriate values in the fields provided and click **OK**.

4. The Fragment Preview dialog opens (see [Section A.66, "Fragment Preview Dialog"](#)), where you can see what your fragment looks like.
5. Click a section in the site hierarchy pane to see how the fragment appears in different sections of the web site.

If a section in the site hierarchy is not included in your site navigation, has no primary page associated with it, or is pointing to an external URL, then you cannot select it and preview it in this dialog.

6. To see what the fragment looks like with different parameter values selected, click **Change Parameter Values**, enter the appropriate values in the fields provided, and then click **OK** to return to the Fragment Preview dialog.
7. Click **Finished** when you're done previewing the fragment.

For a description of all the sample fragments and their parameters in the Toolbox, see [Appendix C, "Sample Fragments."](#)

You cannot preview a static list fragment in the Fragment Preview dialog because the list requires a contributor data file with content in it to be viewed properly.

13.7.2 Choosing Different Parameters for a Fragment

You can customize the way a fragment appears by changing its parameters in the Fragment Parameter Values dialog when you preview a fragment or add it to a page template, subtemplate, or region template. You can also change those values after you've added the fragment.

13.7.3 Choosing Different Parameters When Adding a Fragment

To choose different parameters when adding a fragment, perform these tasks:

1. Place your cursor in the template where you want the fragment to appear.
2. In the Toolbox, click a fragment category: **Navigation Fragments**, **Dynamic List Fragments**, **Static List Fragments**, or **Other Fragments**. Then click the desired fragment.
3. In the Fragment Parameter Values dialog (see [Section A.65, "Fragment Parameter Values Dialog"](#)), enter the appropriate values in the fields provided, and click **OK**.

or

If you're adding a list fragment to a contribution region, enter the appropriate values in the Element dialog and then click **Parameters** to change its parameters.

Click **OK** twice to close the Fragment Parameter Values and the Element dialog.

13.7.4 Changing the Parameters of a Fragment

After you've added a fragment to a template, you can still change its parameters and thus its appearance on a web page.

To change the parameters of a fragment, perform these tasks:

1. In design view, double-click the fragment.

or

In source view, right-click the fragment and choose **Edit Fragment Instance**.

2. Change the desired parameter values in the Fragment Parameter Values dialog (see [Section A.65, "Fragment Parameter Values Dialog"](#)).
3. Click **OK**.

You cannot change the parameter values of a fragment that has been added to a page template if that fragment contains snippets included as type "simple" (see [Section 13.12.4, "Adding, Editing, and Deleting Fragment Snippets"](#)).

For a description of each parameter included in the sample fragments, see [Appendix C, "Sample Fragments."](#) Please note, however, that some fragments may not contain parameters.

To add parameters to a fragment or to further customize the fragment, you can edit the fragment in the Fragment Editor (see [Section 13.8, "Editing Fragments"](#)).

Note: The options you choose in the Fragment Parameter Values dialog apply only to this one instance where you added the fragment to a template. You can choose different values each time you add a fragment to a template without affecting the appearance of the same fragment used elsewhere on your site.

13.8 Editing Fragments

There are two ways to change the appearance of a fragment on your web site: change its parameters (see [Section 13.7.2, "Choosing Different Parameters for a Fragment"](#)) or change the fragment.

By editing a fragment, you can completely change its appearance and functionality to meet your needs. You can even create your own parameters so that you can ensure editing decisions later, when you actually add the fragment to a template.

There are two ways to edit a fragment:

- [Section 13.8.1, "Editing a Copy of a Fragment"](#)
- [Section 13.8.2, "Editing a Fragment"](#)

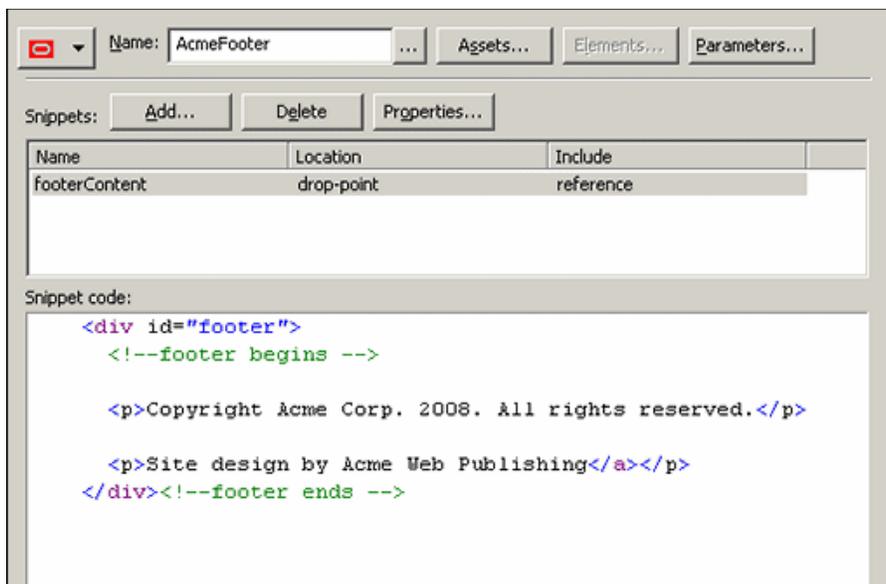
13.8.1 Editing a Copy of a Fragment

Editing a copy of a fragment can be useful when you want to edit a fragment without altering the original. You also must edit a copy of a fragment when the fragment library it belongs to is checked out by another user or labeled as read-only (see [Section 13.13.3, "Making a Fragment Library Read-Only"](#)).

Note: The fragments that come with Oracle Site Studio are read-only, so to edit them, you must choose this option.

To edit a copy of a fragment, perform these tasks:

1. In the Toolbox, click a fragment category: **Navigation Fragments**, **Dynamic List Fragments**, **Static List Fragments**, or **Other Fragments**.
2. Right-click the desired fragment and choose **Copy and Edit**.
3. The Properties for Fragment dialog (see [Section A.58, "Properties for Fragment Dialog"](#)) opens, where you can specify individual properties for the fragment.
4. Click **OK** to close the fragment properties dialog.
5. The fragment opens in the Fragment Editor, where you can start making changes to the fragment (see [Section 13.12, "Using the Fragment Editor"](#)).

Figure 13–2 Fragment Editor with Fragment

- When finished making changes, click the **Save** icon (Figure 13–3) in the toolbar (or click **File** then **Save**) and follow the steps outlined in Section 13.13.2, "Saving a Fragment in a Fragment Library."

Figure 13–3 Save Icon

The fragment displays in the Toolbox with the other fragments, and it is available to every site in the content server.

13.8.2 Editing a Fragment

After you (or another designer) adds a fragment to the Toolbox, you can start editing it right away.

Note: You cannot edit the fragments that come with Oracle Site Studio directly because they are read-only.

To edit a fragment, perform these tasks:

- In the Toolbox, click a fragment category: **Navigation Fragments**, **Dynamic List Fragments**, **Static List Fragments**, or **Other Fragments**.
- Right-click the desired fragment and choose **Edit**.
- The fragment opens in the Fragment Editor, where you can start making changes to the fragment (see Section 13.12, "Using the Fragment Editor").
- When finished, click the **Save** icon (Figure 13–4) in the toolbar (or click **File**, then select **Save**).

Figure 13–4 Save Icon

5. On the Assign Info Form in the content server, enter the appropriate metadata for the library and then click **Assign Info**.

Note: When you edit a fragment in the Fragment Editor, the fragment library (that the fragment is stored in) is checked out of the content server, and it is checked back in when you close the Fragment Editor. (It is also checked out and checked in each time you save the fragment.) This prevents designers (assuming your organization has more than one) from encountering sharing violations or overriding each other's work.

It may occasionally be necessary to click the **Refresh** icon (Figure 13–5) in Designer to see the most current list of fragments in the Toolbox (which reads them from the content server).

Figure 13–5 Refresh Icon



In Oracle Site Studio, it is good practice to create a copy of a fragment and then edit that fragment instead of editing the original. There are several reasons for this:

- You can quickly and easily revert to the original fragment if needed (in case anything goes wrong with the fragment you're editing).
- You can quickly get started with your own fragment without re-creating everything in an existing fragment (much like creating a page template based on an existing page template).
- You reduce the chances of overwriting another designer's work and consequently, creating unpredictable results on a web site. The reason for this is that the fragments you create and edit in Designer are stored in one place in the content server. They appear in every instance of Designer that is connected to the same content server, and therefore, multiple designers can end up using the same fragment on different web sites.

13.9 Creating a New Fragment

In addition to editing fragments and creating fragments from existing ones (see [Section 13.8, "Editing Fragments"](#)), you can create a fragment from scratch and add your own content to it. This is a good choice to introduce functionality or use existing content from another web site (that wasn't created with Oracle Site Studio).

To create a fragment, perform these tasks:

1. From the **File** menu, select **Fragment**, then select **New**.
2. In the New Fragment dialog, enter property values for the fragment (see [Section A.58, "Properties for Fragment Dialog"](#)).
3. Click **OK** to close the fragment properties dialog, which opens the Fragment Editor.
4. In the Fragment Editor, add the desired snippets, assets, parameters, and elements (if it's a static list) to the fragment. For more information, see [Section 13.12, "Using the Fragment Editor"](#)

5. When finished making changes, click the **Save** icon (Figure 13–6) in the toolbar (or click **File** then **Save**) and follow the steps outlined in Section 13.13.2, "Saving a Fragment in a Fragment Library."

Figure 13–6 Save Icon



The fragment displays in the Toolbox with the other fragments and is available to every site on the content server.

13.10 Deleting a Fragment

You can delete a fragment from a template in the same way that you delete tables, graphics, and other objects.

To delete a fragment, perform these tasks:

1. Select the fragment, and press the **Delete** key on your keyboard.

or

In source view, right-click the fragment code and choose **Delete Fragment Instance**.

2. Click **Yes** to confirm the deletion.

The fragment instance is removed. The fragment itself remains in the Toolbox, and you can continue using it on other templates.

13.11 Deleting a Fragment in the Toolbox

You can delete a fragment in the Toolbox, which also deletes it from the content server.

Note: Fragments in Oracle Site Studio are stored in Oracle Content Server and are available to every web site on the server. If you delete a fragment for one web site, other web sites using that same fragment is affected. You should, therefore, first verify that the fragment is not being used elsewhere before you delete it.

To delete a fragment in the Toolbox, perform these tasks:

1. In the Toolbox, right-click the desired fragment.
2. Choose **Delete**.
3. Click **Yes** to confirm the deletion.

The fragment is no longer included in the Toolbox.

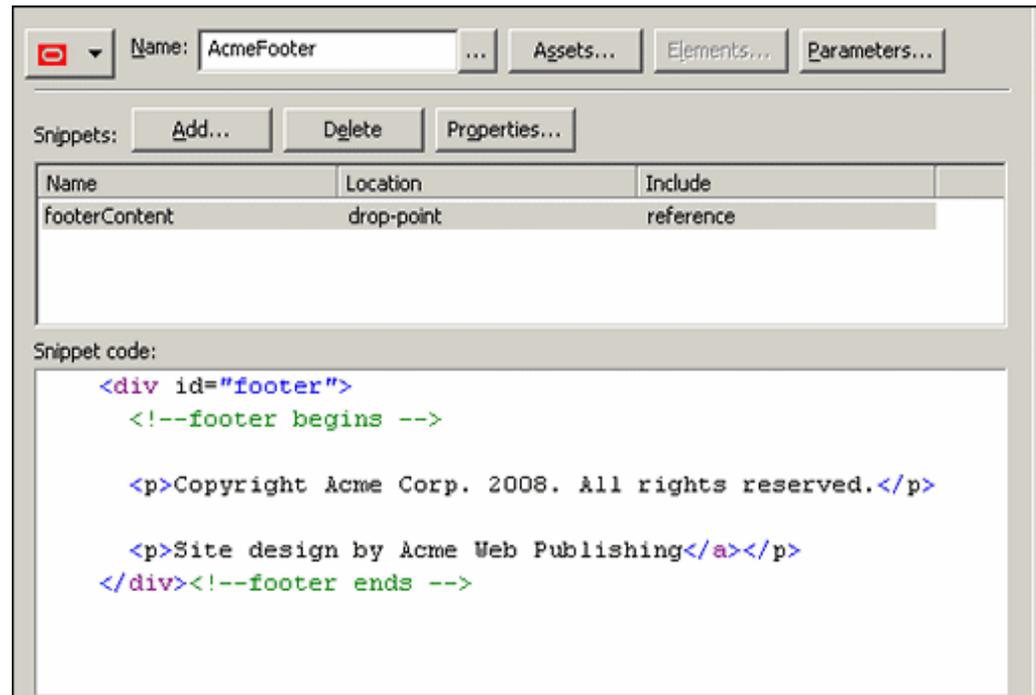
13.12 Using the Fragment Editor

A fragment is a container for code (HTML, XML, JavaScript, Idoc Script), files referenced by that code (graphics, CSS, additional scripts, and so on), and parameters that govern how that code is added to a template.

The file that stores this information is written in XML and is stored in the content server with the fragment assets. The XML file and the fragment assets collectively are called a fragment library.

You can view and edit fragments and the files referenced by that fragment using the Fragment Editor.

Figure 13–7 *Fragment in the Fragment Editor*



Using the Fragment Editor, you can modify existing fragments and create your own fragments. You can turn just about any piece of code or functionality into a fragment that can be added to a template and still managed separately from the template.

The following are some tasks you'll perform in the Fragment Editor:

- [Section 13.12.1, "Opening the Fragment Editor"](#)
- [Section 13.12.2, "Specifying Fragment Properties"](#)
- [Section 13.12.3, "Changing the Icon Associated with a Fragment"](#)
- [Section 13.12.4, "Adding, Editing, and Deleting Fragment Snippets"](#)
- [Section 13.12.5, "Adding, Editing, and Deleting Fragment Assets"](#)
- [Section 13.12.6, "Adding, Editing, and Deleting Fragment Parameters"](#)
- [Section 13.12.7, "Adding, Editing, and Deleting Fragment Elements"](#)

The Preview button in the Fragment Editor performs the same action as using the Preview option in the right-click menu of the Toolbox (see [Section 13.7, "Previewing Fragments"](#)).

13.12.1 Opening the Fragment Editor

The Fragment Editor can be opened within the Designer application, where it used to view and edit fragments. The Fragment Editor displays in the workspace in the same

way that templates display there. In fact, some options in the Designer toolbar also apply to fragments (Cut, Copy, Paste, Save, window controls, and so on).

There are two ways to open the Fragment Editor:

- Right-click an existing fragment in the Toolbox and choose **Edit** or **Copy and Edit** (see [Section 13.8, "Editing Fragments"](#)).

or

- From the **File** menu, select **Fragments**, then select **New** (see [Section 13.9, "Creating a New Fragment"](#)).

13.12.2 Specifying Fragment Properties

Each time you create a fragment in Designer, you are prompted to confirm or create individual properties for the fragment. You can also view and change existing properties from within the Fragment Editor by clicking the **Additional Information** (Figure 13–8) icon:

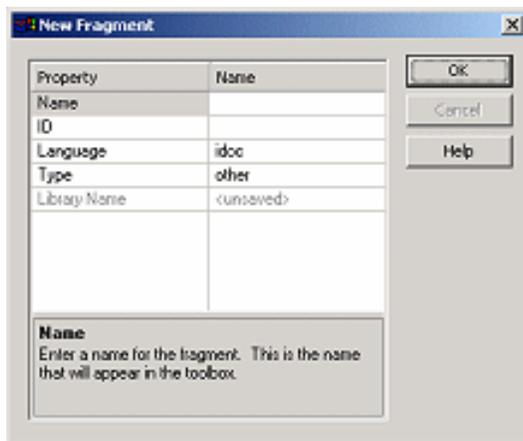
Figure 13–8 *Additional Information icon*



To specify properties for a fragment, perform these tasks:

1. Open the Properties for Fragments dialog (see [Section A.58, "Properties for Fragment Dialog"](#)).

Figure 13–9 *Properties for Fragment Dialog*



2. Enter a name for the fragment in the **Name** field. The name displays in the Toolbox.
3. Enter an identification in the **ID** field. This identifies the fragment and its many parts (snippets, fragment assets, parameters, and so on). The ID also serves as an XML attribute and the base name for the fragment's classes, files, and directories. You cannot change the ID after you assign one to a fragment.

Note: The ID should be concise and should not contain spaces or non-ASCII characters.

4. Choose a language for the fragment in the **Language** field. The language is used for server-side scripting and must correlate with the web site type.
 - Select **idoc** or **jsp** for "hjsp/jsp" sites.
 - Select **asp** for "asp" sites.
5. Choose a fragment type in the **Type** field. The type determines where the fragment appears in the Toolbox (see [Section 13.4, "Fragments in the Toolbox"](#)) and certain editing options available in the Fragment Editor. You have four choices:
 - **navigation**: A navigation fragment provides site navigation for your web site, such as a navigation bar, breadcrumbs, and a search box.
 - **dynamiclist**: A dynamic list fragment provides advanced contribution features by displaying a list of data files or native documents that a contributor can edit.
 - **staticlist**: A static list fragment provides advanced contribution features by displaying a structured layout of elements, whereby a contributor can add, edit, and delete rows and columns of information.
 - **other**: This category of fragments is intended for all other, possibly miscellaneous fragments, such as a copyright, an embedded Flash presentation, and a dynamically converted document.
6. Click **OK** to close the Properties dialog and return to the Fragment Editor.

The **Library Name** in the Properties dialog denotes the fragment library where the fragment is stored. You choose this location when you save the fragment for the first time (see [Section 13.13.2, "Saving a Fragment in a Fragment Library"](#)).

Note: Only ASCII characters should be used for fragment IDs and the names of asset files and folders. This is because the folder within the zip file, which is created automatically for the assets, is named after the fragment ID, and the zip format does not support extended characters. This is true also for the names of any folders used for the assets and for the names of the asset files themselves.

13.12.3 Changing the Icon Associated with a Fragment

Each fragment in the Toolbox has an icon associated with it that can be used to differentiate it from other fragments. You can change this icon to something different, perhaps something that more closely resembles the purpose of the fragment.

To change the icon associated with a fragment, perform these tasks:

1. With the fragment open in the Fragment Editor, click the **Change** icon ([Figure 13–10](#)) in the upper left. (This icon changes with each fragment you select. The icon shown here is the one you see for a new fragment.)

Figure 13–10 *Change Icon*



2. Click the desired icon in the menu.

The icon displays inside this button in the Fragment Editor, and it displays in the Toolbox when you save the fragment.

13.12.4 Adding, Editing, and Deleting Fragment Snippets

A snippet is the actual text or code that a fragment adds to a template. A fragment must have at least one snippet, which is defined by its insertion point (head of page, body, drop-point, and so on) in the template.

For example, if the fragment adds text to one place on the page, then you can create just one snippet for the fragment. But if the fragment inserts text or code in multiple locations on the page (such as a script that must be placed in the head and the body), then you must create two snippets for the fragment (one for the head and one for the body).

You can add, edit, and remove fragment snippets using the Fragment Editor.

To add or edit a snippet, perform these tasks:

1. With the fragment open in the Fragment Editor, click **Add** to create a snippet, or highlight an existing snippet and click **Properties**.

This opens the Snippet Properties dialog (see [Section A.64, "Snippet Properties Dialog"](#)).

2. Enter a name for the snippet in the **Name** field. (The name may contain spaces and special characters, if you like.)
3. Beside **Location**, choose an available option:
 - **drop-point**: Inserts the snippet where the cursor is located on the template.
 - **head**: Inserts the snippet in the head (specifically the end of the <HEAD> tag) of the page template.
 - **top-of-body**: Inserts the snippet at the beginning of the body (immediately following the <BODY> tag) in the page template.
 - **bottom-of-body**: Inserts the snippet at the end of the body (immediately preceding the </BODY> tag) in the page template.
4. Beside **Include**, choose an available option:
 - **simple**: Adds the entire contents of the snippet to the template. If parameters are present, their values are added directly to the template, making them a permanent part of the page. The fragment, consequently, is no longer recognized as a fragment and its parameters cannot be modified (see [Section 13.7.2, "Choosing Different Parameters for a Fragment"](#)).
 - **inline**: Adds the entire contents of the snippet to the template. As an inline snippet, however, the fragment and its parameters are still recognized as a fragment. You can still move the fragment, delete it, and modify its parameters.
 - **reference**: Adds a reference to the snippet in the template. As a referenced snippet, not only is it still recognized and managed as a fragment, but you can still edit the fragment in the Fragment Editor and immediately see your changes on the template (or templates if used more than once). This behavior is similar to that of an include file (HTML, CSS, script, and so on) on other web sites.
5. Beside **Design View Code**, enter the text you would like to appear as a placeholder for the fragment in the design view of a template. (This feature is intended for snippets that are inserted at the drop-point or the body of a page template and not the head.)

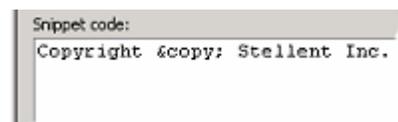
To open the text editor, click the **Additional Information** icon (Figure 13–11), enter your text, and then click **OK** to close the text editor.

Figure 13–11 Additional Information icon



6. Click **OK** to close the snippet properties and return to the Fragment Editor.
7. With the snippet selected, you can add and edit the content of the snippet in the **Snippet code** text box.

Figure 13–12 Snippet Code Text Box



Repeat these steps for every snippet that you would like to add or edit in this fragment. You can add as many snippets as you like.

To delete a snippet, simply highlight that snippet and click **Delete**.

Fragment snippets can be written in HTML, XML, JavaScript, Idoc Script, JavaServer Pages (JSP), and Active Server Pages (ASP). For ASP and JSP, however, you must place your code in a fragment asset (see [Section 13.12.5, "Adding, Editing, and Deleting Fragment Assets"](#)) and then reference that code from an inline fragment snippet.

You should avoid the word "BODY" in an inline head snippet because that name conflicts with code used by Oracle Site Studio to manage the head and body portions of a page template. If you must use this word in your snippet, then you should either (a) change the include method of the snippet from inline to reference, (b) put the code in a fragment asset (see [Section 13.12.5, "Adding, Editing, and Deleting Fragment Assets"](#)), and then include it in your fragment that way, or (c) change the code so that the word "BODY" is split into two parts (two concatenated strings).

If you use the same fragment containing a referenced snippet on multiple web sites on the content server, then any changes you make to that fragment immediately affect all web sites where it is used. If this is not your intention, then you should create different fragments for each site (see [Section 13.8.1, "Editing a Copy of a Fragment"](#)).

Note: With ASP fragments, you can only include a snippet using the "simple" or "inline" method. Including by "reference" is not available because of the way ASP code handles includes.

13.12.5 Adding, Editing, and Deleting Fragment Assets

Assets are the files that are referenced from within a fragment, specifically from within a snippet in the fragment. A graphic, a cascading style sheet, and a standalone JavaScript file (see first note below for implementing ASP or JSP) are all frequently used assets. When you add an asset to a fragment, it becomes a part of the fragment and is managed in a fragment library in the content server. If you move or make a copy of a fragment, all of its assets are included.

You can add, edit, and remove fragment assets in the Fragment Assets dialog in the Fragment Editor.

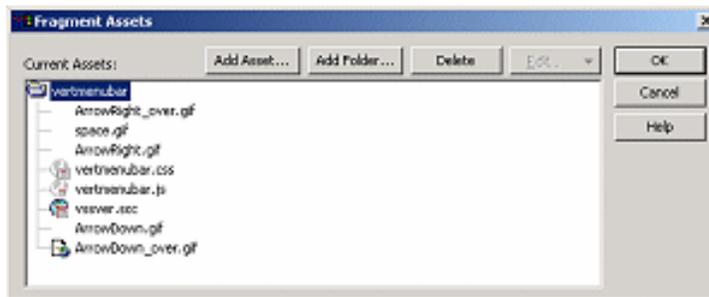
- [Section 13.12.5.1, "Adding an Asset"](#)
- [Section 13.12.5.2, "Editing an Asset"](#)
- [Section 13.12.5.3, "Deleting an Asset"](#)

13.12.5.1 Adding an Asset

To add an asset, perform these tasks:

1. With the fragment open in the Fragment Editor, click **Assets**.
This opens the Fragment Assets dialog (see [Section A.59, "Fragment Assets Dialog"](#)).

Figure 13–13 *Fragment Assets Dialog*



2. To create a subfolder for the asset (this is optional), click **Add Folder**, type a name for the folder, and then press **Enter** on your keyboard.
By default, Oracle Site Studio creates an asset folder based on the fragment's ID (see [Section 13.12.2, "Specifying Fragment Properties"](#)). The folder name uses all lowercase letters to ensure that the fragment works on all platforms (specifically, UNIX). If you create any subfolders, you should use all lowercase letters as well. In addition, you should use only ASCII characters (see note below).
3. Highlight the folder that you would like to add the asset to and then click **Add Asset** (the root, or top-level folder, is the default location for assets).
4. Browse to the file on your file system and then click **Open** in the Windows dialog.
5. Click **OK** to close the Fragment Assets dialog and return to the Fragment Editor.

When you save your fragment, the asset is packaged with the fragment and stored in the fragment library on the content server. You can now refer to the asset from each snippet in the fragment.

13.12.5.2 Editing an Asset

To edit an asset, perform these tasks:

1. With the fragment open in the Fragment Editor, click **Assets**.
2. In the Fragment Assets dialog, select the asset and then click **Edit** to open the two edit options.

Figure 13-14 Edit Options in Fragment Assets Dialog

- Click **Edit** to open the asset in the default application associated with that file type (for example, a cascading style sheet may open in Notepad).
 - Click **Edit with** to open the Edit With dialog where you can use a different application to edit the file.
3. Make your changes to the asset and then close the application used to edit the asset.
 4. Click **OK** to close the Fragment Assets dialog and return to the Fragment Editor.

13.12.5.3 Deleting an Asset

To delete an asset, perform these tasks:

1. With the fragment open in the Fragment Editor, click **Assets**.
2. In the Fragment Assets dialog, select the asset and then click **Delete**.
3. Click **OK** to close the Fragment Assets dialog and return to the Fragment Editor.

Only ASCII characters should be used for fragment IDs and the names of asset files and folders. This is because the folder within the zip file, which is created automatically for the assets, is named after the fragment ID, and the zip format does not support extended characters. This is true also for the names of any folders used for the assets and for the names of the asset files themselves.

Rather than edit a cascading style sheet (CSS) asset that is included in a fragment, you can just as well change the CSS declarations in the fragment snippet to reference your own cascading style sheet.

We recommend that you use the same name for your text-based assets (cascading style sheets, JavaScript files, and so on) as you are using for the fragment ID (see [Section 13.12.2, "Specifying Fragment Properties"](#)). For example, if the fragment ID is "myfragment," then you should use the names "myfragment.css," "myfragment.js," and so on. Not only does this help you easily identify the files associated with a fragment, but if you ever create a fragment based on this one (using the "Copy and Edit" feature), all of the assets and code within them are updated to reflect the fragment name. As such, the assets seamlessly work with the fragment.

Note: If you plan to write your fragments in JSP or ASP, you should place your code in a fragment asset and then reference that code from an inline fragment snippet.

13.12.6 Adding, Editing, and Deleting Fragment Parameters

Parameters are a set of attributes that govern the appearance and behavior of a fragment when it is added to a template. When a fragment contains parameters, you are prompted to enter information in the Fragment Parameter Values dialog before adding the fragment to a template. Parameters can be used to add such things as text, color, size, or a query in the content server. And you can change the value of the parameters each time you add the fragment to a template.

While parameters are optional when building a fragment, they are very useful to make a fragment customizable and reusable. You can add most of a fragment's functionality to the source of the fragment, or you can add it as a parameter that can be changed when the fragment is actually used.

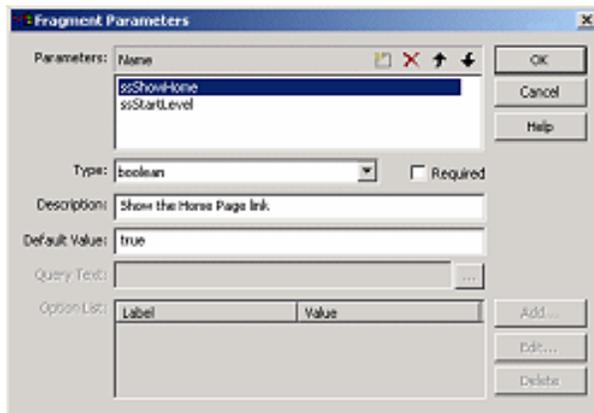
When parameters are used, you can have several renditions of the same fragment on the same template, where different parameters are used in each instance. Moreover, parameters can be useful for developers who have written code for the web site and would like to hand off that code as a packaged feature (a customizable fragment) to a site designer. (This scenario assumes there are multiple users in Designer.)

13.12.6.1 Editing the Parameter in a Fragment

To edit the parameter in a fragment, perform these tasks:

1. With the fragment open in the Fragment Editor, click **Parameters**.
2. In the Fragment Parameters dialog (see [Section A.58, "Properties for Fragment Dialog"](#)), select a parameter in the **Name** list.

Figure 13–15 Parameter Selection in Fragment Parameters Dialog



3. Make the appropriate changes to the **Type**, **Description**, **Default Value**, **Query Text**, and **Option List** (Query Text and Option List do not apply to all parameter types). For more information, see [Section A.58, "Properties for Fragment Dialog."](#)
4. Click the **Parameter Up** icon ([Figure 13–16](#)) or the **Parameter Down** icon ([Figure 13–17](#)) to move the parameter up or down in the list (this also affects its location in the Fragment Parameter Values dialog).

Figure 13–16 Parameter Up icon



Figure 13–17 Parameter Down icon



5. Click **OK** to close the Fragment Parameters dialog.

To see a description of the parameters in the sample fragments, see [Appendix C, "Sample Fragments."](#)

13.12.6.2 Adding a New Parameter to a Fragment

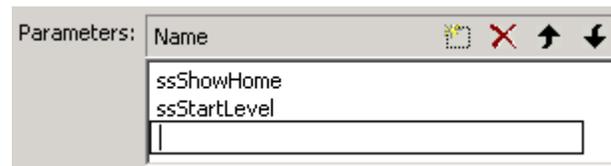
To add a parameter to a fragment, perform these tasks:

1. With the fragment open in the Fragment Editor, click **Parameters**.
2. In the Fragment Parameters dialog (see [Section A.58, "Properties for Fragment Dialog"](#)), click the **New Parameter** icon ([Figure 13–18](#)) and type a name for the parameter in the text box provided.

Figure 13–18 *New Parameter Icon*



Figure 13–19 *Parameter Name Entry in Fragment Parameters Dialog*



3. Choose a type from the **Type** menu (for a description of each type, see [Section A.61, "Fragment Parameters Dialog"](#)).
4. Enter a description in the **Description** field.
The description displays at the bottom of the Fragment Parameter Values dialog when you use the fragment.
5. Enter a value for the **Default Value**.
This value is used if you do not choose a parameter value (in the Fragment Parameter Values dialog) when adding the fragment to a template.
6. Enter a value for **Query Text** and **Option List** (this applies only to parameters of type "text," "manageddoc," and "managedurl").
7. Click the **Parameter Up** icon ([Figure 13–20](#)) or the **Parameter Down** icon ([Figure 13–21](#)) to move the parameter up or down in the list (this also affects its location in the Fragment Parameter Values dialog).

Figure 13–20 *Parameter Up icon*



Figure 13–21 *Parameter Down icon*



8. Click **OK** to close the Fragment Parameters dialog.

13.12.6.3 Inserting Parameters and Parameter Declarations in a Snippet

After you create one or more parameters in the Fragment Parameters dialog, you can insert those parameters into the individual snippets in the fragment (for more information on snippets, see [Section 13.12.4, "Adding, Editing, and Deleting Fragment Snippets"](#)).

The Fragment Editor includes a convenient option to insert parameters and parameter declarations directly into your snippet code. To insert them, simply place your cursor

in the code where you would like to add the parameter or parameter declaration, right-click, and then choose the appropriate action.

Figure 13–22 Menu for Inserting a Parameter Declaration

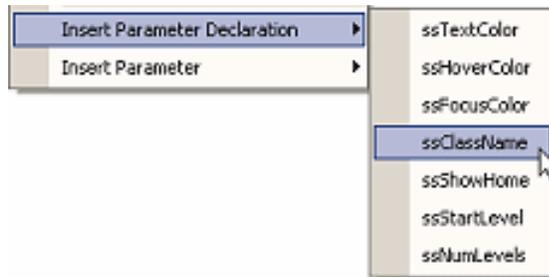
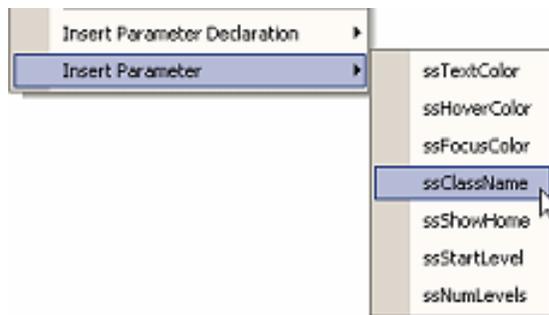


Figure 13–23 Menu for Inserting a Parameter



A good way to learn how to create and use parameters is to open a sample fragment from the Toolbox and see how its parameters are used. For a description of these fragments and their parameters, see [Appendix C, "Sample Fragments."](#)

13.12.7 Adding, Editing, and Deleting Fragment Elements

Elements are areas in a contribution region that can be edited using the Contributor application. These elements appear as fields in Contributor, where users (contributors) can edit text, graphics, and other content.

The most common use of an element is to first create a contribution region on a page template, and then add one or more elements (WYSIWYG, plain text, image, or custom) to that region. Contributors can then open the web page containing the region and add and edit the text or graphics in each field.

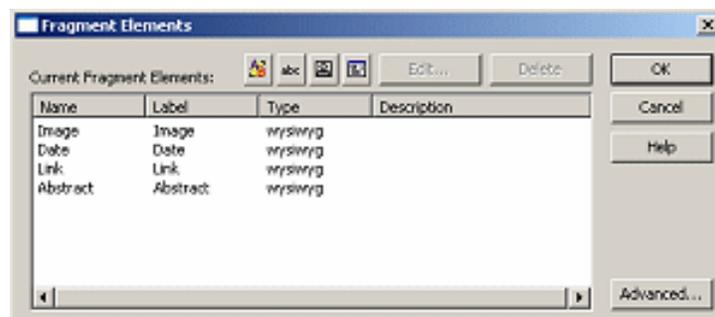
A more advanced use of an element is to include it in a fragment to create a static list. A static list bundles multiple elements in a table-like layout that allows contributors to add, edit, delete, and even rearrange rows and columns of elements (see [Section C.3, "Static List Fragments"](#)).

13.12.7.1 Adding an Element to a Fragment

To add an element to a fragment, perform these tasks:

1. With the fragment open in the Fragment Editor, click **Elements**.

This opens the Fragments Elements dialog (see [Section A.60, "Fragment Elements Dialog"](#)).

Figure 13–24 *Fragment Elements Dialog*

2. Click the type of element to add to the fragment: WYSIWYG, plain text, image, or custom.

Figure 13–25 *WYSIWYG Element Icon***Figure 13–26** *Plain Text Element Icon***Figure 13–27** *Image Element Icon***Figure 13–28** *CustomElement Icon*

3. In the Element dialog, follow the same steps as if you were adding the element to a contribution region.
4. Click **OK** to close the Element dialog and then **OK** again to close the Fragment Elements dialog.

13.12.7.2 Editing an Element in a Fragment

To edit an element in a fragment, perform these tasks:

1. With the fragment open in the Fragment Editor, click **Elements**.
2. In the Fragment Elements dialog (see [Section A.60, "Fragment Elements Dialog"](#)), select the desired element and then click **Edit**.
3. In the Element dialog, follow the same steps as if you were editing an element in a contribution region.
4. Click **OK** to close the Element dialog and then **OK** again to close the Fragment Elements dialog.

13.12.7.3 Deleting an Element in a Fragment

To delete an element in a fragment, perform these tasks:

1. With the fragment open in the Fragment Editor, click **Elements**.

2. In the Fragments Element dialog (see [Section A.60, "Fragment Elements Dialog"](#)), select the desired element and then click **Delete**.
3. Click **OK** to close the Fragment Elements dialog.

The Elements button only displays in the Fragment Editor when the fragment is of the type "staticlist" (see [Section 13.12.2, "Specifying Fragment Properties"](#)).

The Fragment Elements dialog contains an **Advanced** button. This opens a text editor that you can use to customize your element even more. However, when you start editing an element this way, you cannot return to the Fragment Elements dialog graphical user interface because there is no interface support for these edits.

Static list fragments are designed for use with the Contributor application. As such, when you add the static list to a page template, you must add it to a contribution region and then assign a data file to that region.

13.13 Fragment Libraries

A fragment is a container for code (HTML, XML, JavaScript, Idoc Script, and optionally JSP or ASP), files referenced by that code (graphics, CSS, and additional scripts), and parameters that govern how that code is added to a template.

A fragment library is a container for one or more fragments, and it is how Oracle Site Studio stores and manages fragments.

You work with fragment libraries in the following ways:

- [Section 13.13.1, "About Fragment Libraries"](#)
- [Section 13.13.2, "Saving a Fragment in a Fragment Library"](#)
- [Section 13.13.3, "Making a Fragment Library Read-Only"](#)
- [Section 13.13.4, "Uploading and Downloading Fragment Libraries"](#)

13.13.1 About Fragment Libraries

Fragments are stored in fragment libraries. A fragment library is stored in the content server as a managed content item.

The library comprises two parts: an XML file and a zip file. The XML file describes every fragment in the library, and the zip file holds all of the fragment assets referenced by each fragment in the library.

In the content server, the XML file serves as the alternate file (the Web Location), and the zip file serves as the native file. If you were to search for the fragment library "SS_Fragments_Examples" (a library included with Oracle Site Studio), you would see the following on the content information page:

Metadata	Assigned values
Content ID:	SS_Fragments_Examples
Title:	SS_Fragments_Examples
WebsiteObjectType:	Fragment
Web Location:	HTTP://<Site>/groups/public/documents/adacct/ss_fragments_examples.xml
Native File:	SS_Fragments_Examples.zip

The default fragments that ship with Oracle Site Studio are stored in the following fragment libraries:

- SS_Fragments_IDOC
- SS_Fragments_JSP
- SS_Fragments_ASP
- SS_Fragments_Examples
- SS_Fragments_Dynamic
- SS_Fragments_Plain
- SS_Fragments_CSP_Examples

13.13.2 Saving a Fragment in a Fragment Library

When you save a fragment that you've created with the Fragment Editor, you must choose a fragment library to store the fragment in. You can store the fragment in its own library, or you can store it, along with several others, in the same library. You are prompted for this the first time you save the fragment.

To save a fragment in a fragment library, perform these tasks:

1. When finished making changes to a fragment (see [Section 13.8, "Editing Fragments"](#)), click the **Save** icon ([Figure 13–29](#)) on the Designer toolbar (or click **File** then select **Save**).

Figure 13–29 Save Icon



The Save Fragment dialog (see [Section A.67, "Save Fragment Dialog"](#)) opens.

2. Click **New** to create a library for this fragment.
Or, if you have created a library, select it in the list and then click **OK**.
3. On the Assign Info Form, enter the appropriate metadata for the library (Content ID, Type, Title, and so on) and then click **Assign Info**.

The library is created in the content server, and it displays in the Save Fragment dialog (preceded with the text "server:") the next time you open it.

Note: The Save Fragment dialog opens only the first time you save a fragment. You're taken directly to the Assign Info Form with each subsequent save.

As you create and manage fragments, you may find it easier to group related fragments in relatively small numbers (or in some cases put a single fragment in a library by itself) because fragments cannot be moved around, backed up, or distributed individually. They are handled and managed as fragment libraries (see [Section 13.13.4, "Uploading and Downloading Fragment Libraries"](#)).

The default fragments that ship with Oracle Site Studio are stored in read-only fragment libraries, which is why you don't see them in the Save Fragment dialog (and also why you cannot edit the individual fragments).

You can make your own fragment libraries read-only to prevent others from accidentally overwriting or erasing a fragment (see [Section 13.13.3, "Making a Fragment Library Read-Only"](#)).

13.13.3 Making a Fragment Library Read-Only

Making a fragment library read-only can be very useful, especially if you're working with multiple designers in the same organization. The sample fragments that come with Oracle Site Studio are all stored in read-only fragment libraries to prevent accidental erasure or overwriting.

By making your fragment library read-only:

- The fragments in the fragment library cannot be overwritten or deleted by others (they can only be copied and edited).
- The fragment libraries won't appear in the Save Fragment dialog, which prevents others from adding fragments to that library.

To make a fragment read-only, perform these tasks:

1. Log onto the content server and search for the fragment library.
2. Open the Content Information Page for the fragment library.
3. Right-click the Web Location file (with the .xml file extension), choose **Save Target As**, and save it to your system.
4. Check the fragment library out of the content server and download the native file (with the .zip extension).
5. Open the .xml file in a text editor and add the text **readonly="true"** to the root element <fragments> tag. The resulting tag should look like this:

```
<fragments name="SS_Fragments" readonly="true">
```

6. Save and close the file and then check both files back into the content server, specifying the .zip file as the **Primary File** and the .xml file as the **Alternate File**.

It may be necessary to refresh the Toolbox in Designer to see the read-only fragments.

As an alternative to these steps, you could also use Designer's Download Fragment Library feature, extract the .xml file out of the downloaded zip, re-archive the zip, and use Designer's Upload Fragment Library feature to return the library to the content server (see [Section 13.13.4, "Uploading and Downloading Fragment Libraries"](#)).

13.13.4 Uploading and Downloading Fragment Libraries

After you create and edit a fragment or collection of fragments, you can move the fragment library (which the fragments are stored in) from one content server to another. This can be especially useful if you're working with other designers or developers and want to share fragments with them. Or, perhaps you simply want to create a backup of your fragment library for safekeeping.

To do this, you use the Download Fragment Library and Upload Fragment Library feature in Designer.

13.13.4.1 Downloading a Fragment Library

When you download a fragment library, Oracle Site Studio places the library (currently located in the content server) in a single zip file and downloads it onto your system. You can then easily transport the zip file.

To download a fragment library from the server, perform these tasks:

1. Click **File** then **Fragments**, then select **Download Fragment Library**.
2. In the Oracle Content Server Dialog, click **Select** beside the fragment library to download.
3. In the Windows Save As dialog, browse to a location on your file system and then click **Save**.

13.13.4.2 Uploading a Fragment Library

When you upload a fragment library (after downloading it following the previous steps), Oracle Site Studio checks the zip file into the content server and extract its contents to the appropriate directory (\weblayout\fragments\) where it can be used by the web site.

To upload a fragment to the server, perform these tasks:

1. Click **File** then **Fragments**, then select **Upload Fragment Library**.
2. In the Windows Open dialog, browse to the fragment library (zip file) on your file system and then click **Open**.
3. On the Assign Info Form, enter a Content ID, Type, and Title for the fragment library.

Note: Do not alter the WebsiteObjectType metadata. It must be set to "Fragment" to be recognized by your web site.

4. Click **Assign Info** to close the Oracle Content Server Dialog and return to Designer.

Instead of using the Upload Fragment Library feature in Designer, you can also check the fragment library into the content server directly and then use the Oracle Site Studio Administration page to deploy the fragment library to the required location. For more information on Fragment Library deployment, see *Managing Oracle Site Studio*.

Working With Site Reports

Oracle Site Studio provides two types of report that you can view: web site objects reports and site usage reports.

This section covers the following topics:

- [Section 14.1, "About Site Reports"](#)
- [Section 14.2, "Web Site Objects Report"](#)
- [Section 14.3, "Site Usage Report"](#)
- [Section 14.4, "Viewing a Web Site Objects Report"](#)
- [Section 14.5, "Viewing a Web Site Usage Report"](#)

14.1 About Site Reports

When managing large web sites or managing multiple sites, there are times when you want to know how many files are associated with the site, where those files are used, and how often those files are viewed by certain users. You may, for example, want to know where a particular data file is located on the site and how often it is being viewed. The same goes for native documents, fragments, scripts, and more.

One way to do this is to use the site reports feature in Oracle Site Studio. You can view the site reports online or print them out, if you like.

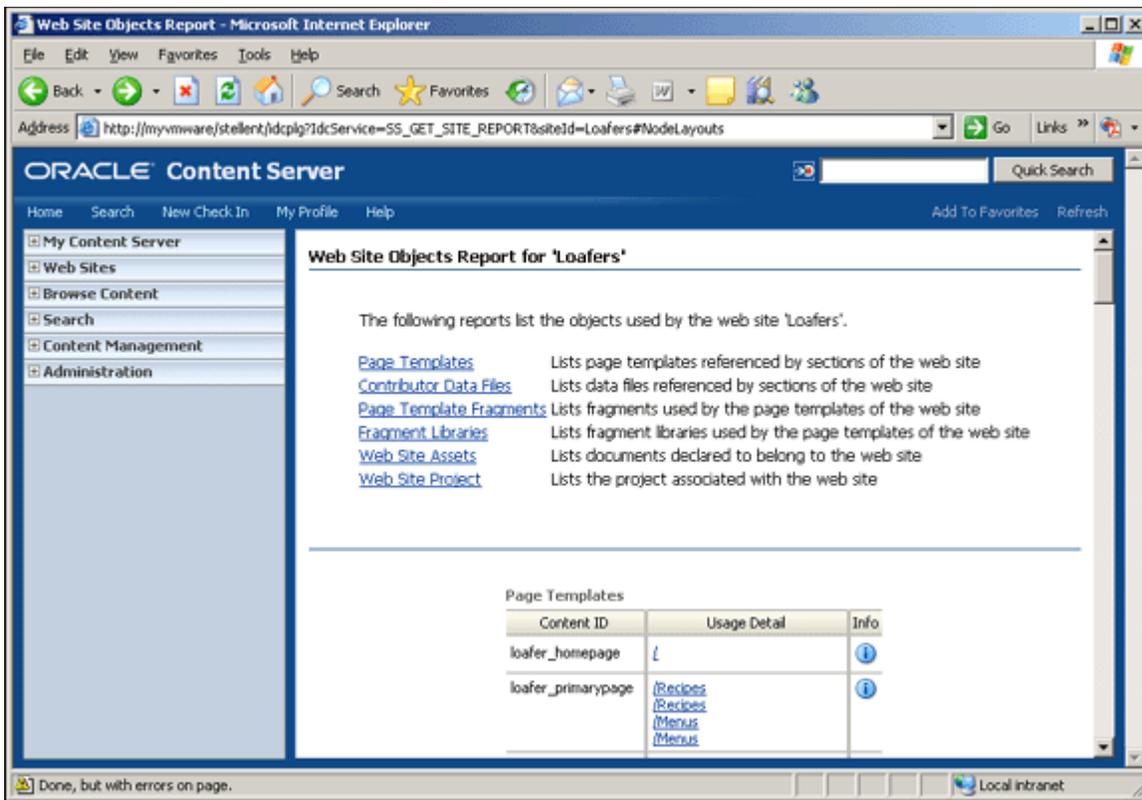
There are three types of reports that you can view:

- [Section 14.2, "Web Site Objects Report"](#)
- [Section 14.3, "Site Usage Report"](#)

14.2 Web Site Objects Report

The web site objects report ([Figure 14–1](#)) shows all of the explicitly referenced files used by your site. This report displays files by category (Page Templates, Contributor Data Files, Fragment Libraries, and so on).

Figure 14–1 Web Site Objects Report

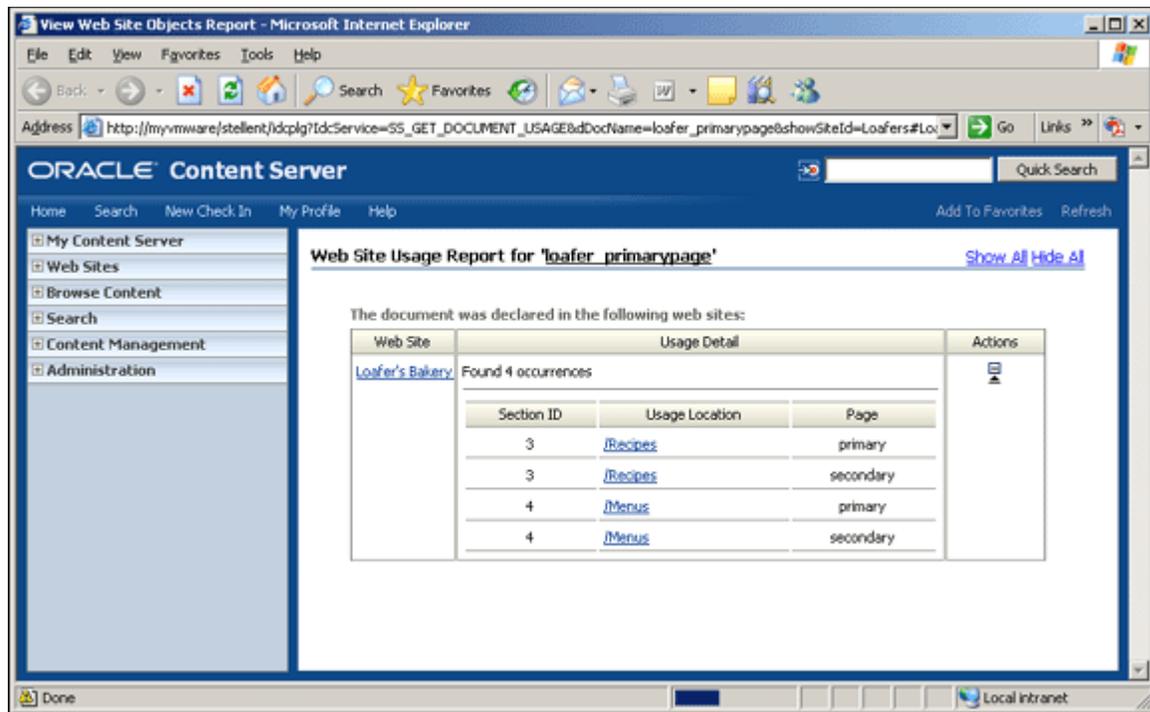


For more information, see [Section 14.4, "Viewing a Web Site Objects Report."](#)

14.3 Site Usage Report

The site usage report (Figure 14–2) shows a detailed summary of a single file, specifically where it is being used. The report breaks this down by web sites where it is used, sections where it is used, and so on.

Figure 14–2 Web Site Usage Report Detailing a Page Template



For more information, see [Section 14.5, "Viewing a Web Site Usage Report."](#)

Important: One of the criteria used to generate a report is the metadata that is assigned to each file. Therefore, if a file contains incorrect metadata, it may improperly appear in a report or not appear at all. The more efficient your metadata model, the more efficient the site reports are.

14.4 Viewing a Web Site Objects Report

A site objects report shows you all of the files used by your site. When you view the report, you see a breakdown of files by category (Page Templates, Contributor Data Files, Fragment Libraries, and so on).

You can view the report from various places in Oracle Site Studio.

- [Section 14.4.1, "Viewing the Report in Designer"](#)
- [Section 14.4.2, "Viewing the Report on the Oracle Site Studio Administration Page"](#)

14.4.1 Viewing the Report in Designer

To view the report in Designer, perform one of these tasks:

- Click **File** then select **Site**, then select **Advanced** then **View Web Site Objects Report**.
- or*
- In the Site Hierarchy pane, right-click the root of the web site and choose **View Web Site Objects Report**.

14.4.2 Viewing the Report on the Oracle Site Studio Administration Page

To view the report using the Oracle Site Studio Administration page, perform these tasks:

1. Open the Oracle Site Studio Administration page.
2. Click **Manage Web Sites**.
3. Highlight the web site that you would like to view a report on.
4. Click **View Web Site Objects Report**.

You can preview the files used on your site by clicking the links, where available.

Click the **Home Page** icon (Figure 14–3) to view the home page of your site.

Figure 14–3 Home Page icon



Click the **Information** icon (Figure 14–4) to view the content information page for a file.

Figure 14–4 Information icon



In order for a file to appear in this report, it must contain the required metadata that matches your site.

14.5 Viewing a Web Site Usage Report

A site usage report provides a detailed summary of a single file, specifically where it is being used on one or more sites in the content server. The report itself is stored in the content server. When you view the report, you see a breakdown of the web sites where the file is used, the sections where it is used, and so on.

You can view the report from various places in Oracle Site Studio:

- [Section 14.5.1, "Viewing a Report for a Page Template in Designer"](#)
- [Section 14.5.2, "Viewing a Report for a Data File or Native Document in Designer"](#)
- [Section 14.5.3, "Viewing a Report for a Data File or Native Document From the Web Page"](#)
- [Section 14.5.4, "Viewing a Report for All Other Files"](#)

14.5.1 Viewing a Report for a Page Template in Designer

To view a report for a page template in Designer, perform these tasks:

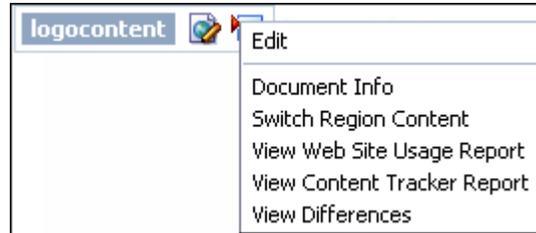
1. In the Site Hierarchy pane, expand the section containing the page template to view a report for.
2. Right-click the page template (primary or secondary) and choose **View Web Site Usage Report**.

14.5.2 Viewing a Report for a Data File or Native Document in Designer

To view a report for a data file or native document in Designer, perform these tasks:

1. Open the Assign Region Content dialog in Designer (see [Section A.54, "Assign Content Dialog"](#)).
2. Open the menu from the **Contribution** icon ([Figure 14-5](#)).

Figure 14-5 Contribution icon menu



3. Click **View Web Site Usage Report**.

14.5.3 Viewing a Report for a Data File or Native Document From the Web Page

To view a report for a contributor data file or native document from the web page, perform these tasks:

1. Browse to the web page where the data file or native document appears.
2. Enable contribution mode.
3. Open the menu from the **Contribution** icon and choose **View Web Site Usage Report**.

This option is only available if it is enabled in the placeholder definition in Designer.

14.5.4 Viewing a Report for All Other Files

To view a report for all other files, perform these tasks:

1. Open the Content Information Page for the file to view a report for.
2. Using the menu beside Actions, choose **View Web Site Usage Report**.

You can preview the places where the file is used on your site by clicking the links, where available.

Click the **Home Page** icon ([Figure 14-6](#)) to view the home page of your site.

Figure 14-6 Home Page icon



Click **Show All** to show the details of where the file is used on every web site, or click **Hide All** to hide these details. Click the **[+]** icon to show the details of where the file is used on a given site, or click the **[-]** icon to hide these details.

In order for a file to appear in the report, it must have the required metadata matching your site.

You can also view site usage reports in Contributor (see *Using Oracle Site Studio Contributor*).

Oracle Content Server's workflow features can be used with Oracle Site Studio to create and maintain your site. With workflows, you can ensure a smooth flow of information from the designer to the manager to the contributor, and so on. You can also control what passes between designers, such as a page template, or a cascading style sheet.

This section covers the following topics:

- [Section 15.1, "About Workflows"](#)
- [Section 15.2, "Workflow Types"](#)
- [Section 15.3, "Workflow Experience for Contributors"](#)
- [Section 15.4, "Setting Up a Workflow"](#)
- [Section 15.5, "Workflow Examples"](#)

For more information on workflows, see *Using Oracle WebCenter Content*.

15.1 About Workflows

When you design a workflow for Oracle Site Studio, for the most part you follow the same processes and steps that you would if you were creating a workflow for any other content item on the content server. One key difference is that you must determine which contribution regions (that is, editable parts of a web page) are part of the workflow and who has access to content in those regions.

For more information on how a contributor gains access to a workflow to review a portion of a web site, see [Section 15.3, "Workflow Experience for Contributors."](#)

15.2 Workflow Types

There are two types of workflows:

- **Basic workflows**, in which the review process is initiated manually for specific content items.
- **Criteria workflows**, in which a content item enters the workflow automatically when it is checked into the content server and its metadata matches predefined criteria specified in Oracle Site Studio Designer and the content server.

Criteria workflows are most commonly used with Oracle Site Studio.

15.3 Workflow Experience for Contributors

To add or edit content on a web site, a contributor typically clicks the **Edit** icon in the contribution graphic on a web page in contribution mode to open Oracle Site Studio Contributor, where the content assigned to the contribution region is available in an editable form. With workflows, a participant first receives an e-mail message with a link to the web page to be reviewed. Clicking this link opens the web page already in contribution mode.

After the web page is in contribution mode, one or more contribution graphics are visible, one for each editable contribution region on the page. If the content of a contribution region is in a workflow, the contribution graphic includes a special icon to the left of the region name (a green gear) to indicate this (see [Figure 15-1](#)).

Figure 15-1 Contribution Graphic Showing Workflow Indicator



Contributor graphic, with the workflow icon on the left to indicate that the editable contribution region is currently in a workflow.

The contributor can then choose a workflow option in the contribution graphic menu, depending on how the workflow was designed and the contributor's role in it:

- **Approve Document:** If the contributor clicks this option, the next participant in the workflow is notified automatically by e-mail. After the last workflow participant accepts the contributor data file that stores the content for the web page, the contributor data file is released to its final destination. The web page itself can then become "live" on the web site.
- **Reject Document:** If the contributor clicks this option, they have the opportunity to complete a rejection form to explain why they are rejecting the content. An e-mail is then sent to the previous reviewer in the workflow.
- **Edit:** Clicking this option in the menu is the same as clicking the contribution icon without workflow set up. That is, Contributor opens with the content of the contribution region in editable form.

15.4 Setting Up a Workflow

Setting up a workflow in Oracle Site Studio is the same as setting up a workflow in Oracle Content Server, with one additional step: when you set up a placeholder definition in Designer, you must enable workflow functionality for it. This adds workflow options to the placeholder when the page is viewed in contribution mode.

To enable workflow functionality for a placeholder definition, select the appropriate workflow checkboxes:

- **Workflow approve:** Select this checkbox to add an "Approve Document" option to the contribution graphic menu, allowing the reviewers in a workflow to approve the region content.
- **Workflow reject:** Select this checkbox to add a "Reject Document" option to the contribution graphic menu, allowing the reviewers in a workflow to reject the region content.

15.5 Workflow Examples

It may be useful to use workflow functionality for any the following content items used by Oracle Site Studio:

- [Section 15.5.1, "Contributor Data Files in Workflows"](#)
- [Section 15.5.2, "Native Documents in Workflows"](#)
- [Section 15.5.3, "Images in Workflows"](#)

These files must contain the appropriate Oracle Site Studio metadata to be recognized as part of the web site. You may decide to enter any one of these content types into workflow, perhaps using a custom metadata field of "workflow" with values that are recognizable by your organization.

Note: Page templates, subtemplates, region templates, and scripts do not have the opportunity for approval/rejection within workflow if they are edited with Designer.

15.5.1 Contributor Data Files in Workflows

Forcing a contributor data file to go through a criteria workflow is the most common workflow process for contributors to the web site. This file represents the actual content of a web page and is therefore updated with information whenever a contributor edits the web page in Contributor.

When you add a contributor data file to a workflow, you are in effect ensuring that any changes contributors make to a contribution region of a web site go through an approval process before the changes can be posted to the live web site.

15.5.2 Native Documents in Workflows

Adding a native document to a criteria workflow could be useful in several situations:

- Inline dynamic conversions
- Random dynamic conversions
- Dynamic lists that use dynamically converted documents

For native document to be converted to HTML successfully, Dynamic Converter must be installed and running on the content server.

Inline Dynamic Conversions

It may be useful to add a native document to a criteria workflow if you have an inline dynamic conversion (where the document is included in the site layout) to ensure that any changes to the native document are subject to an approval process before the document is actually included in the web site.

Random Dynamic Conversions

It may be useful to add a native document to a criteria workflow if you have a random dynamic conversion (in which a random dynamic converter fragment containing a query is added to the web site) to ensure that any content items checked into the content server do not become part of the site without first going through an approval process.

Note: A content item is included in the content server search results when it is released from a workflow. As a result of this, a content item appears in a dynamic list only if there is a released revision of that item.

For example, you could display a randomly selected quote on your web site through dynamic conversion where every native document added with a quote is included in the site in no particular order. If you place these documents into a criteria workflow, a document is not released until it has gone through the workflow and has been approved. It is then included in the dynamic conversion results and displayed on the web site.

Dynamic Lists That Use Dynamically Converted Documents

It may be useful to add a native document to a criteria workflow if you have a dynamic list that targets dynamically converted documents. This enables you to ensure that all native documents checked into the content server that match the dynamic query results go through a workflow before they are added to the dynamic list that opens on the web site. This approach could be useful to display a dynamic list of press releases on your web site that targets dynamically converted documents.

Note: A content item is included in the content server search results when it is released from a workflow. As a result of this, a content item appears in a dynamic list only if there is a released revision of that item.

15.5.3 Images in Workflows

Adding an image to a criteria workflow may be useful to ensure, for example, that all images checked into the content server are reviewed and approved by your graphic designer before they are added to the live web site.

Setting Up Manager

Oracle Site Studio Manager is a web-based application that can be used to modify the site hierarchy and make changes to each section. The users (site managers) use it to add and remove sections, assign page templates, change site properties, and more.

This section covers the following topics:

- [Section 16.1, "About Oracle Site Studio Manager"](#)
- [Section 16.2, "Choosing Where and When to Display Manager"](#)
- [Section 16.3, "When Manager Opens"](#)
- [Section 16.4, "Adding Manager to Your Site"](#)
- [Section 16.5, "Creating a Manager Configuration Settings File"](#)
- [Section 16.6, "Copying a Manager Configuration Settings File"](#)
- [Section 16.7, "Customizing Manager Configuration Settings"](#)
- [Section 16.8, "Viewing the Content Information of a Manager Configuration Settings File"](#)
- [Section 16.9, "Adding a Manager Configuration Settings File to a Site"](#)
- [Section 16.10, "Removing a Manager Configuration Settings File from a Site"](#)
- [Section 16.11, "Description of the Manager Configuration Settings File"](#)

16.1 About Oracle Site Studio Manager

Oracle Site Studio Manager is a web-based application that can be used to modify the site hierarchy and make changes to each section. The users (site managers) use it to add and remove sections, assign page templates, change site properties, and more.

Manager is similar to Contributor in that the user opens the application directly from the site and starts making changes. It is added to the site differently than Contributor. There are no contribution regions; the designer simply adds the Oracle Site Studio Manager fragment from the Toolbox. To customize the appearance of Manager, you modify the fragment. To customize the functionality of Manager, you edit the manager configuration settings file (see [Section 16.7, "Customizing Manager Configuration Settings"](#)).

Note: For more information about using Oracle Site Studio Manager, see *Managing Oracle Site Studio*.

16.2 Choosing Where and When to Display Manager

Choosing where and when to display Manager on the site is an important decision because it affects how site managers use the site. Unlike Contributor, there is no immediate relationship with the content using a contribution icon. Instead, Manager is used to change settings that affect the entire page or site (like assigning page templates and changing section properties).

As such, you want to create a logical location for Manager and choose the appropriate time to display it (always on or only display in contribution mode).

There are three places where Manager can display:

- **From every page on the site:** You can add Manager to every page on the site to give users a sense they are modifying the section or page they are viewing.
- **From a designated section:** You can add Manager to a designated section of the site where users go to manage everything. This may give users the sense they are managing the site from a central location.
- **From the Oracle Site Studio Administration page:** Manager is always accessible from the Manage Web Sites page on the content server. Any customizations you make to Manager do not affect this page.

If you add Manager to a section, you probably want to make that section viewable only in Contributor. You might also spend time designing a page template specifically for Manager.

16.3 When Manager Opens

There are two states when Manager opens:

- **Contribution mode:** You can have Manager display only in contribution mode so that the general public does not see it and instead only those with contribution access do.
- **All times:** You can have Manager display on the site at all times so that everyone, including all public users, sees it.

The default setting in Manager is to display at all times.

16.4 Adding Manager to Your Site

When you add Manager to your site, you allow a user (a site manager) to modify the site hierarchy and perform several site management tasks. You can add Manager to your site in two ways:

- [Section 16.4.1, "Adding Manager With Default Settings"](#)
- [Section 16.4.2, "Adding Manager With Custom Settings"](#)

16.4.1 Adding Manager With Default Settings

To add Manager to your site, simply add the Oracle Site Studio Manager fragment to a page template. When you do this, you're adding Manager with the maximum amount of settings enabled for the user.

To add Manager with its default settings, perform these tasks:

1. Open a page template from the site hierarchy and place your cursor where you want Manager to appear on the page.

2. In the Toolbox, click **Other Fragments**.
3. Click the **Oracle Site Studio Manager** fragment.
The Fragment Parameter Values dialog opens (see [Section A.65, "Fragment Parameter Values Dialog"](#)).
4. Choose a theme: **blue**, **red**, or **grey**.
To change the look and feel beyond what is available in these themes, edit the Oracle Site Studio Manager fragment using the Fragment Editor (see [Section 13.8, "Editing Fragments"](#)).
5. Click **OK** to close the Fragment Parameter Values dialog.
The Manager application now appears on the page when viewed in a browser.

16.4.2 Adding Manager With Custom Settings

If you want to change the settings that are available to the user, then you should first create a manager configuration settings file, edit the file, and then add the Oracle Site Studio Manager fragment, but this time, reference the manager configuration settings file you created.

To add Manager with a manager configuration settings file, perform these tasks:

1. Follow the steps to create a manager configuration settings file (see [Section 16.5, "Creating a Manager Configuration Settings File"](#)).
2. Open a page template from the site hierarchy and place your cursor where you want Manager to appear on the page.
3. In the Toolbox, click **Other Fragments**.
4. Click the **Oracle Site Studio Manager** fragment.
The Fragment Parameter Values dialog opens (see [Section A.65, "Fragment Parameter Values Dialog"](#)).
5. Choose a theme: **blue**, **red**, or **grey**.
To change the look and feel beyond what is available in these themes, edit the Oracle Site Studio Manager fragment using the Fragment Editor (see [Section 13.8, "Editing Fragments"](#)).
6. Place the cursor in the **Name** column of the settings fields, and click the icon ([Figure 16–1](#)) next to it.

Figure 16–1 Additional Information Icon



The search results page opens, showing all available manager configuration settings files on the content server.

7. Click **Select** beside the manager configuration settings file you want to use.
8. Click **OK** to close the Fragment Parameter Values dialog.

The Manager application now appears on your page template with the options that you enabled in the manager configuration settings file.

16.5 Creating a Manager Configuration Settings File

A manager configuration settings file is an editable XML file read by Manager to determine the number of editing options the site manager has. If you're content with the default options turned on in Manager, then you can simply add the Oracle Site Studio Manager fragment without creating a manager configuration settings file.

If, however, you want to control which options are enabled and disabled then you must create a manager configuration settings file, which you do in the Site Assets pane. There are two starting points of manager configuration settings files you can create: one with the minimal settings enabled and one with the maximum settings enabled. You can create as many manager configuration settings files as you like.

To create a manager configuration settings file, perform these tasks:

1. Open the Site Assets pane in Designer.
2. Click the menu at the top and choose **Manager Configuration Settings**.
3. Click the **New File** icon (Figure 16–2).

Figure 16–2 New File Icon



4. To create a manager configuration settings file, perform one of the following:
 - Choose **New** and then **Minimal Enabled** to create a Manager interface with the minimum amount of user settings.
 - Choose **New** and then **Maximum Enabled** to create a Manager interface with the maximum number of user settings.

For more information, see [Section 16.11, "Description of the Manager Configuration Settings File."](#)

Or, to create a copy of an existing manager configuration settings file:

- Choose **Copy** and then **Selected** to copy the currently selected manager configuration settings file to a new one.
- Choose **Copy** and then **from Server** to copy an existing file on the content server.
- Choose **Copy** and then **from Local** to copy an existing file on your file system.

The standard content check-in page opens.

5. Provide the metadata for the manager configuration settings file and click **Assign Info** at the bottom of the page to check it into the content server. All required metadata fields are marked red.

The manager configuration settings file appears in the Site Assets pane, where you can select if for editing or reference it from a fragment. For information on customizing the settings file, see [Section 16.7, "Customizing Manager Configuration Settings."](#)

16.6 Copying a Manager Configuration Settings File

You can select a manager configuration settings file from the list to copy, select a manager configuration setting from the content server to copy, or select a manager configuration setting from your local server to copy.

Copying a manager configuration settings file from the list

1. In the menu on the Site Assets pane, select **Manager Configuration Settings**.
2. Select the manager configuration settings file you want to copy from the list.
3. Click the **Create New** icon (Figure 16–3), select **Copy**, and then **Selected**:
The Assign Info Form opens for you to check the item into the content server.

Figure 16–3 Create New Icon

4. Enter appropriate values for the Assign Info Form.
5. When complete, click **Assign Info**.
6. The manager configuration setting is copied.

Copying a manager configuration settings file from the server

1. In the menu on the Site Assets pane, select **Manager Configuration Settings**.
2. Click the **Create New** icon (Figure 16–3), select **Copy**, and then **from Server**.
A search results page opens.
3. Click the corresponding **Select** button of the ones you want to copy.
The Assign Info Form opens for you to check the item into the content server.
4. Enter appropriate values for the Assign Info Form.
5. When complete, click **Assign Info**.
6. The manager configuration settings file is copied.

Copying a manager configuration settings file from your local instance

1. In the menu on the Site Assets pane, select **Manager Configuration Settings**.
2. Click the **Create New** icon (Figure 16–3), select **Copy**, and then **from Local**.
A navigation window opens.
3. On your local instance, navigate to the manager configuration settings file you want to copy.
4. Select the manager configuration setting and click **Open**.
The Assign Info Form opens for you to check the item into the content server.
5. Enter appropriate values for the Assign Info Form.
6. When complete, click **Assign Info**.
7. The manager configuration settings file is copied.

16.7 Customizing Manager Configuration Settings

To customize the available options in Manager, you edit the manager configuration settings file. This is an XML file read by Manager to determine the available options to the user (a site manager). When you modify this file, you can control whether users can add sections, assign page templates, modify site properties, and so on.

You edit a manager configuration settings file using the Site Assets pane:

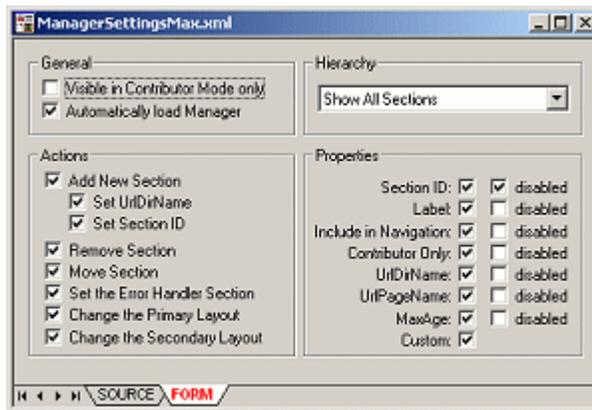
1. In the Site Assets pane, choose the asset type **Manager Configuration Settings**.
2. Select the desired manager configuration settings file and click the **Edit** icon (Figure 16-4).

Figure 16-4 Edit Icon



The Manager Configuration Settings dialog [Section A.69, "Manager Configuration Settings Dialog"](#) opens in form view (Figure 16-5).

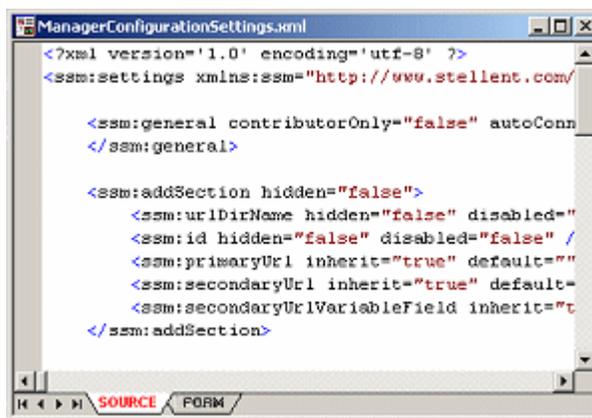
Figure 16-5 Form View of Manager Configuration Settings File



Note: Many of the settings are disabled (grayed-out) if the manager configuration settings file you are editing has the hierarchy hidden.

3. Select or clear the checkbox next to each setting enabled or disabled for the user. For more information on each setting, see [Section 16.11, "Description of the Manager Configuration Settings File."](#)
4. Alternatively, click the **Source** tab to edit the XML source code of the manager configuration settings file directly:

Figure 16-6 Editing Manager Configuration File Settings in Source View



5. For more information, see [Section 16.11, "Description of the Manager Configuration Settings File."](#) Also, see *Technical Reference for Oracle Site Studio* for details on the source code, tags, attributes, and the like.
6. Save your work and close the manager configuration settings file.

To change the appearance of Manager, including the color scheme, fonts, and other layout settings, edit the Oracle Site Studio Manager fragment in the Fragment Editor (see [Section 13.8, "Editing Fragments"](#)).

16.8 Viewing the Content Information of a Manager Configuration Settings File

To view content information for a manager configuration settings file, perform these tasks:

1. In the menu on the Site Assets pane, select **Manager Configuration Settings**.
2. Select a manager configuration setting from the list.
3. Click the **Doc Info** icon ([Figure 16-7](#)).

The content information page opens.

Figure 16-7 Doc Info icon



16.9 Adding a Manager Configuration Settings File to a Site

To add a manager configuration settings file to a site, perform these tasks:

1. In the menu on the Site Assets pane, select **Manager Configuration Settings**.
2. Click the **Add to Site** icon ([Figure 16-8](#)).

A search results page opens.

Figure 16-8 Add to Site Icon



3. Select the manager configuration setting to add, click **Oracle Site Studio**, and then **Select Marked Documents**.
4. Depending on your configuration, you may receive a caution that you are about to add existing asset(s) to your site. Click **OK**.

The manager configuration setting is now associated with the web site, and you can now open it to edit.

16.10 Removing a Manager Configuration Settings File from a Site

To remove a manager configuration settings file from a site, perform these tasks:

1. In the menu on the Site Assets pane, select **Manager Configuration Settings**.
2. From the list, select the manager configuration setting to be removed and click the **Remove From Site** icon ([Figure 16-9](#)).

When you remove a manager configuration setting, you are simply removing it from the site, not deleting from the content server.

Figure 16–9 Remove from Site Icon



16.11 Description of the Manager Configuration Settings File

A manager configuration settings file is read by the Manager application to determine the available editing options for the user (the site manager). The options include the ability to add and remove sections, assign page templates, and change section properties. As the designer of the site, you can enable and disable each option.

Here is a description of each setting in the manager configuration settings file when viewed in form view:

Setting	Description
Visible in Contributor Mode only	Manager opens only when viewed in contribution mode.
Automatically load Manager	Manager automatically opens on the page when you view it in a browser.
Hierarchy	Only these portions of the site hierarchy can be modified by the user. Choose one of three options: Hide: Users cannot modify the site hierarchy at all. Show All Sections: Users can modify the entire site hierarchy. Show Current Section Only: Users can modify only the current section (the one they are viewing Manager from).
Add New Section	Users can add a section to the site hierarchy and specify the following information when creating a section. Set UrlDirName: The URL that represents the section in the web site address. Set Section ID: The ID of the section used by Oracle Site Studio to call up a section of the site.
Remove Section	Users can remove a section from the site hierarchy.
Move Section	Users can move a section in the site hierarchy.
Set the Error Handler Section	Users can designate the section as an error handler (for use with a custom error page).
Change the Primary Page Template	Users can change the primary page template associated with the section.
Change the Secondary Page Template	Users can change the secondary page template associated with the section.
Section ID	Users can change the ID associated with the section. disabled: Displays the value but prevents users from changing it.
Label	Users can change the label associated with the section. disabled: Displays the value but prevents users from changing it.

Setting	Description
Include in Navigation	Users can control whether this section appears in the site navigation. disabled: Displays the value but prevents users from changing it.
Contributor Only	Users can control whether this section appears only when the site is viewed in contribution mode. disabled: Displays the value but prevents users from changing it.
UrlDirName	Users can specify the URL that represents the section in the web site address. disabled: Displays the value but prevents users from changing it.
UrlPageName	Users can specify the URL that represents the primary page in this section. If a name is not specified (by you or the site manager), then "index.htm" is used. disabled: Displays the value but prevents users from changing it.
MaxAge	Users can specify the maximum amount of time that the page can be cached, which is particularly useful when the site is using a reverse proxy cache for delivery. disabled: Displays the value but prevents users from changing it.
Custom	Users can change any custom properties that have been created for the site.

User Interface

Oracle Site Studio Designer includes numerous dialogs and administrative pages that you use to create and administer your site:

- [Section A.1, "Customize Dialog"](#)
- [Section A.2, "Site Connection Manager Dialog"](#)
- [Section A.3, "Site Connection Details Dialog"](#)
- [Section A.4, "Create New Site Dialog"](#)
- [Section A.5, "Site Addresses Dialog"](#)
- [Section A.6, "Choose Default Link Format Dialog"](#)
- [Section A.7, "Define Environment Properties Dialog"](#)
- [Section A.8, "New Section Dialog"](#)
- [Section A.9, "Custom Section Properties Dialog"](#)
- [Section A.10, "Define Custom Section Property Dialog"](#)
- [Section A.11, "Select \(Home\) Page Template Dialog"](#)
- [Section A.12, "Select \(Primary\) Page Template Dialog"](#)
- [Section A.13, "Select \(Secondary\) Page Template Dialog"](#)
- [Section A.14, "Site Asset Categories Dialog"](#)
- [Section A.15, "Enable Metadata Modification Dialog"](#)
- [Section A.16, "Select Site Dialog"](#)
- [Section A.17, "Select Section Dialog"](#)
- [Section A.18, "Link Wizard"](#)
- [Section A.19, "Contribution Region Dialog"](#)
- [Section A.20, "Region Content Options Dialog"](#)
- [Section A.21, "Choose Document Types Dialog"](#)
- [Section A.22, "Edit Query Text Dialog"](#)
- [Section A.23, "Conversion Definition Dialog"](#)
- [Section A.24, "Native Document Conversion Settings Dialog \(Legacy\)"](#)
- [Section A.25, "Native Document Conversion Settings Dialog"](#)
- [Section A.26, "WYSIWYG Element Dialog"](#)

-
- Section A.27, "WYSIWYG Element Definition Dialog"
 - Section A.28, "Plain Text Element Dialog"
 - Section A.29, "Plain Text Element Definition Dialog"
 - Section A.30, "Image Element Dialog"
 - Section A.31, "Image Element Definition Dialog"
 - Section A.32, "Custom Element Dialog"
 - Section A.33, "Custom Element Definition Dialog"
 - Section A.34, "Custom Element Settings Dialog"
 - Section A.35, "Static List Element Dialog"
 - Section A.36, "Static List Element Definition Dialog"
 - Section A.37, "Elements Dialog"
 - Section A.38, "Static List Element Validation Dialog"
 - Section A.39, "Dynamic List Element Dialog"
 - Section A.40, "Dynamic List Element Definition Dialog"
 - Section A.41, "Dynamic List Settings Dialog"
 - Section A.42, "Region Definition Dialog"
 - Section A.43, "Element Dialog"
 - Section A.44, "Link Settings Dialog"
 - Section A.45, "Select CSS Classes Dialog"
 - Section A.46, "Select a CSS StyleSheet Dialog"
 - Section A.47, "Choose Fragment Asset Dialog"
 - Section A.48, "WYSIWYG Element Validation Dialog"
 - Section A.49, "Plain Text Element Validation Dialog"
 - Section A.50, "Select HTML Tags Dialog"
 - Section A.51, "Image Element Validation Dialog"
 - Section A.52, "Advanced Element Validation Dialog"
 - Section A.53, "New Item Settings Dialog"
 - Section A.54, "Assign Content Dialog"
 - Section A.55, "Choose Region Content Dialog"
 - Section A.56, "Generate Unique Region Content Dialog"
 - Section A.57, "Fragment Editor Dialog"
 - Section A.58, "Properties for Fragment Dialog"
 - Section A.59, "Fragment Assets Dialog"
 - Section A.60, "Fragment Elements Dialog"
 - Section A.61, "Fragment Parameters Dialog"
 - Section A.62, "Edit Query Text Dialog"
 - Section A.63, "New Option Dialog"

- Section A.64, "Snippet Properties Dialog"
- Section A.65, "Fragment Parameter Values Dialog"
- Section A.66, "Fragment Preview Dialog"
- Section A.67, "Save Fragment Dialog"
- Section A.68, "Manage Site Page"
- Section A.69, "Manager Configuration Settings Dialog"
- Section A.70, "Placeholder Section Properties Dialog"
- Section A.71, "Edit Placeholder Section Properties Dialog"
- Section A.72, "Placeholder Definition Dialog"
- Section A.73, "Placeholder Definition Mappings Dialog"
- Section A.74, "Define Placeholder Definition Mapping Dialog"
- Section A.75, "Define Placeholder Dialog"
- Section A.76, "Switch Content Wizard"
- Section A.77, "Switch Region Template Dialog"
- Section A.78, "Insert Dynamic Conversion Dialog"
- Section A.79, "Custom Configuration Scripts Dialog"
- Section A.80, "Contribution Element Dialog"
- Section A.81, "Select Fragment Dialog"
- Section A.82, "Oracle Site Studio Replicator"

A.1 Customize Dialog

The Customize Dialog is used to modify the designer interface to match the needs of the designer.

- Section A.1.1, "Customize Dialog: Commands Tab"
- Section A.1.2, "Customize Dialog: Toolbars Tab"
- Section A.1.3, "Customize Dialog: Tools Tab"
- Section A.1.4, "Customize Dialog: Keyboard Tab"
- Section A.1.5, "Customize Dialog: Menu Tab"
- Section A.1.6, "Customize Dialog: Options Tab"
- Section A.1.7, "Customize Dialog: Source View Tab"
- Section A.1.8, "Customize Dialog: Warning Dialogs Tab"
- Section A.1.9, "Customize Dialog: Miscellaneous Tab"
- Section A.1.10, "Customize Dialog: Log File Tab"

A.1.1 Customize Dialog: Commands Tab

The Designer application includes numerous menus and commands. You can change the way these menus and commands appear and even create menus and then populate them with commands. In conjunction with the Toolbars tab (see [Section A.1.2, "Customize Dialog: Toolbars Tab"](#)), you can create a toolbar to display frequently used

commands.

Element	Description
Categories	Displays existing menus. Use New Menu to create a menu in Designer.
Commands	Displays available commands. Use New Menu to create a command for a menu in Designer.
Description	Displays a description of the selected menu.
Help	Opens the online help for this specific dialog.
Close	Closes the Customize dialog.

A.1.2 Customize Dialog: Toolbars Tab

The Designer application includes numerous toolbars. You can show and hide these toolbars, create a toolbar and then populate it with commands (see [Section A.1.1, "Customize Dialog: Commands Tab"](#)); delete toolbars; and restore icons that were removed from the original toolbars.

Element	Description
Toolbars	<p>Displays a checkbox for each toolbar item in Designer. Select the checkbox to show that item in the toolbar or clear the checkbox to hide that item in the toolbar.</p> <p>Contributor Toolbar: Includes Insert Region, Insert Element, and so on.</p> <p>Formatting Toolbar: Includes typeface, bold, align left, and so on.</p> <p>HTML Toolbar: Includes HTML styles, insert image, insert rule, and so on.</p> <p>Menu Bar: Includes File, Edit, View, and so on.</p> <p>Site Toolbar: Includes Choose Site, Assign Content, and so on.</p> <p>Standard: Includes Copy, Paste, Save, and so on.</p> <p>Table Toolbar: Includes Insert Table, Insert Row, Insert Column, and so on.</p>
Reset	<p>Resets the existing toolbar. Select the toolbar you want to restore and then click this button.</p> <p>This command applies only to the default set of toolbars provided with Designer. If you created a toolbar and then removed an icon from it but wanted to restore that icon, then you must go to the Commands tab, drag the icon from the Commands box, and drop it onto the toolbar.</p>
Reset All	<p>Resets all of the toolbars in Designer.</p> <p>This command applies only to the default set of toolbars provided with Designer. If you created a toolbar and then removed an icon from it but wanted to restore that icon, then you must go to the Commands tab, drag the icon from the Commands box, and drop it onto the toolbar.</p>
New	<p>Opens the Toolbar Name dialog where you can create a toolbar.</p> <p>When the toolbar displays in the Toolbars column, drag its icon from the Toolbars tab to the desired position in relation to the other toolbars in Designer.</p> <p>To populate the toolbar, open the Commands tab and drag each icon (representing a command) from the Commands box to the toolbar. You can further customize the toolbar by right-clicking the icon or text and clicking Button Appearance in the popup menu.</p>
Rename	<p>Renames a toolbar. You can only rename toolbars that you create. You cannot rename the toolbars that come with Designer.</p>
Delete	<p>Deletes a toolbar. You can only delete toolbars that you create. You cannot delete the toolbars that come with Designer.</p>

Element	Description
Show text labels	Displays explanatory text with the toolbar icon.
Help	Opens the online help for this specific dialog.
Close	Closes the Customize dialog.

A.1.3 Customize Dialog: Tools Tab

The Designer application includes numerous toolbars and menus. In addition to these, you can add a shortcut to an outside application in the Tools menu to provide immediate access to that application.

Element	Description
Menu contents	Displays the names of existing menus.
	Creates the menu. A placeholder for the tool appears in the Menu contents pane.
	Deletes the menu that you created.
	Moves the item up in the Tools menu.
	Moves the item down in the Tools menu.
Command	Specifies the path to the executable file (the application) that is used for this menu. Use the Browse button to browse to the executable file rather than typing in the full path. You can also type the name of the executable file without the path and then use the initial directory to identify the path.
Arguments	Use to direct the application to open a particular file. The argument requires the full path and file name for the executable (used to open the file) followed by the full path and file name of the file. The path for the executable should have forward slashes; there should be a space between the two paths; and the path off the document should be enclosed in quotation marks. For example: <code>C:/Program Files/Office/winword.exe "C:\Documents\sa.doc"</code>
Initial directory	The directory for the executable file only if the executable file by itself was entered in the Command field; otherwise, this field can be left empty.
Help	Opens the online help for this specific dialog.
Close	Closes the Customize dialog.

A.1.4 Customize Dialog: Keyboard Tab

The Designer application includes numerous commands that have one or more keyboard shortcuts assigned to them. You can create, change, and delete keyboard shortcuts.

Element	Description
Category	Displays the menus. Select the menu that contains the command you want to assign a shortcut to.
Commands	Displays the available commands for the menus. Select the command to assign the shortcut to.

Element	Description
Description	Displays a description of the command.
Set Accelerator for	Specifies when the keyboard shortcut is available. Select Default if you want the keyboard shortcut to be available in general when working in Designer.
Current Keys	Displays keyboard shortcuts already assigned to the command.
Press New Shortcut Key	Place your cursor in this text box then, using your keyboard, press the actual key combination to assign to the shortcut.
Assign	Assigns the shortcut key to the command.
Remove	Deletes the keyboard shortcut.
Reset All	Restores all of the keyboard shortcuts available when you first started Designer.
Help	Opens the online help for this specific dialog.
Close	Closes the Customize dialog.

A.1.5 Customize Dialog: Menu Tab

The Designer application includes numerous popup menus. You can customize the behavior of these popup menus by adding, deleting, and changing the way the popup menus work.

Element	Description
Show Menu for	Selects the menu to modify. Designer uses the Default Menu.
Reset	Resets the menu for Designer.
Menu animations	Controls how the menu commands appear in Designer (includes menus on the Menu bar and popup menus). There are four options: None: The list of menu commands appears in the usual way. Unfold: The list of menu commands opens from top-left to bottom-right. Slide: The list of menu commands opens from top to bottom. Fade: The list of menu commands slowly appears on-screen.
Menu shadows	Displays a shadow effect around the menu.
Select context menu	Modifies the menu command. When you choose this, a popup menu appears in the upper-left corner of the Customize dialog. Click the Commands tab and under Categories, select the appropriate menu. Under Commands, select the command to add to the popup menu. Drag the command onto the popup menu and position it where you would like it to appear on the menu.
Reset	Resets the commands from the Select Context Menu drop-down list.
Help	Opens the online help for this specific dialog.
Close	Closes the Customize dialog.

A.1.6 Customize Dialog: Options Tab

The Designer application includes numerous menus, icons, and screen tips. You can control the way these work on this tab.

Element	Description
Show ScreenTips on toolbars	Opens popup text (a screentip) when the cursor passes over an icon.
Show shortcut keys in ScreenTips	The screentip that opens when the cursor passes over an icon also displays the shortcut key assigned to that command.
Large Icons	Displays large icons in the user interface.
Menus show recently used commands first	Commands are listed based on their frequency of use.
Show full menus after a short delay	Opens the full menu after a short delay.
Reset my usage data	Returns the Designer to its original state.
Help	Opens the online help for this specific dialog.
Close	Closes the Customize dialog.

A.1.7 Customize Dialog: Source View Tab

The Designer application opens multiple views of site assets. These can include Source view, Design view, and Preview. You can customize the way the code opens in source view by changing the font face, font size, and certain automatic formatting.

Element	Description
Font	Selects the font (from those available on your computer) that is used to display the code in Source view.
Size	Selects the font size that is used to display the code in Source view.
Apply automatic formatting when switching to Source view	Each time you switch to Source view when working on an asset, the code is automatically formatted according to the following choices.
Indent block level tags	Successive block-level tags are indented to make the text easier to read.
Indent Size	Sets the indent size by number of characters.
Place each tag attribute on a new line	Each attribute of a tag appears on a new line. The attributes are indented to the same degree as the tag itself.
Tab size	Identifies the number of characters per tab.
Enable Line Wrapping	Enables line wrapping.
Line Length	Specifies the maximum number of characters per line when you enable line wrapping.
Treat unknown tags as inline elements	Ensures that any unknown tags that Designer encounters are treated as generic inline elements (for example, SPAN) during HTML Tidy actions.
Generate XHTML-compliant markup	Turns existing HTML into XHTML-compliant markup. Consequently, elements in HTML are combined with XML to create a more standards-aware language (for example, closing all HTML tags).
doctype	Adds the specified doctype to the header of the HTML for the web page.
Help	Opens the online help for this specific dialog.
Close	Closes the Customize dialog.

A.1.8 Customize Dialog: Warning Dialogs Tab

The Designer application opens numerous warning and confirmation messages. You can control which warning messages open and which ones remain hidden.

Element	Description
Enable Warning Dialogs	Provides checkboxes beside each warning message. Select the checkbox to display the warning message, or clear the checkbox to hide the warning message. By default, certain warning messages open when you first open Designer.
Check All	Checks all of the warning messages so that all of them open.
Clear All	Clears all of the warning messages so that none of them open.
Help	Opens the online help for this specific dialog.
Close	Closes the Customize dialog.

A.1.9 Customize Dialog: Miscellaneous Tab

When Designer opens, it automatically connects to the last site you worked on and it downloads certain fragments for display in the Toolbox. You can change these options, if you like.

Element	Description
Filter fragments in other languages	When checked, only the fragments written in the same language as your site (HCSP/JSP or ASP) appear in the Toolbox.
Filter fragments from other sites	Only the fragments that belong to your site (and not other sites in the content server) appear in the Toolbox. Fragments in fragment libraries not identified as part of your site no longer appear in the Toolbox. To make a fragment part of your site, add it as a site asset to the "Fragment Libraries" category.
Filter default libraries	Only the fragments that you have created for your site (and not the default ones that ship with Oracle Site Studio) appear in the Toolbox.
Reconnect at startup	When opening Designer, you are automatically connected to the last site you worked on.
Enable project status checking	While working in Designer, a project status icon displays in the lower right telling you if the site hierarchy matches the latest project file in the content server.
Enable accessibility mode	While working in Designer, the menus are changed to work better with a screen reader.
Help	Opens the online help for this specific dialog.
Close	Closes the Customize dialog.

A.1.10 Customize Dialog: Log File Tab

Oracle Site Studio creates a detailed log file for your site. You can control what is logged and the details of the log on this tab.

Element	Description
Reset log file each time Designer starts	Creates a log file each time Designer starts.

Element	Description
Log details of all communication with the Content Server	Creates a detailed log of all communication with the content server.
Help	Opens the online help for this specific dialog.
Close	Closes the Customize dialog.

A.2 Site Connection Manager Dialog

Designer uses a site connection to connect to and update a web site in the content server. You can create, edit, and delete site connections using this dialog.

Element	Description
Connection Name web site ID Server Cgi URL (Status area)	Displays each connection name, the corresponding web site, and content server Cgi URL.
Automatically connect when done	Automatically connects to the site (using the site connection you just created) when you close the Site Connection Manager dialog.
New	Opens the Site Connection Details dialog (see Section A.3, "Site Connection Details Dialog"), where you can create a site connection.
Edit	Opens the Site Connection Details dialog (see Section A.3, "Site Connection Details Dialog"), where you can edit an existing site connection.
Delete	Deletes an existing site connection.
Done	Closes the Site Connection Manager dialog and connects to the web site using the highlighted site connection (if you checked "Automatically connect when done").
Help	Opens the online help for this specific dialog.

You must first disconnect from a web site before you can edit or delete that site in the Site Connection Manager.

A.3 Site Connection Details Dialog

Designer uses a site connection to connect to and update a web site in the content server. You can create and edit site connections using this dialog. When creating a site connection, you identify the content server where the site resides, the name of the actual web site, and a name for the connection.

Element	Description
Server Cgi URL	The address of the content server that hosts the web site. You can type the complete Cgi URL address of the content server, or you can choose a placeholder address in the menu, replacing <server> with the name of your server.
Connect	Connects to the content server using the Cgi URL address you provided above. You are prompted to enter your login credentials to connect to the server.

Element	Description
Site Label	Displays existing web sites on the server. Choose an existing site or click New to create a new one.
New	Opens the Create New Site dialog (see Section A.4, "Create New Site Dialog"), where you can create a web site.
Connection Name	The name that displays in Designer to identify this site connection. You can use the default name provided or enter a new one.
OK	Saves your settings and closes the Site Connection Details dialog.
Cancel	Cancels your settings and closes the Site Connection Details dialog.
Help	Opens the online help for this specific dialog.

A.4 Create New Site Dialog

Designer uses a site connection to connect to and update a web site in the content server. You can create a web site using this dialog (accessible from the Site Connection Details dialog (see [Section A.3, "Site Connection Details Dialog"](#))). When creating a site, you specify a site ID, name, and type.

Element	Description
ID	The identification of the web site. This ID is used to uniquely identify this site and its associated files in the content server (not to be confused with a Content ID). You should use only valid characters (no spaces, no symbols, and so on), which this dialog forces you to do when you enter a value. You may also want to keep the value relatively short. You see the ID again when working with the "siteid" fragment parameter or CSP.
Name	The name of the web site. The name that you use becomes the title of the project file in the content server, and it appears as the title of your site in the "Web Sites" menu in the content server. (This value does not affect the TITLE tag in your page templates.)
Type	The type, or scripting choice, for your web site: hcsp/jsp : Creates an Idoc Script or JavaServer Pages-enabled web site, which governs other choices in Oracle Site Studio. asp : Creates an Active Server Pages-enabled web site, which governs other choices in Oracle Site Studio. You can create a web site in HCSP, JSP, or ASP. But only those sites built in HCSP have the full functionality, architecture and features introduced in Oracle Site Studio 10gR4.
OK	Saves your settings and closes the Create New Site dialog.
Cancel	Cancels your settings and closes the Create New Site dialog.

A.5 Site Addresses Dialog

You can open your site using a domain-based address or a folder-based address. Using this dialog, you can add one or more domain names that point to your web site.

Element	Description
	Click this to add a domain name that resolves to your web site. (When you click this, a text box opens where you can enter your address and then click Enter .)
	Deletes the selected site address.
Set Default	Sets the selected address as the default address. The default address is used in several places in Oracle Site Studio, such as by Designer to connect to and preview your site and when you assign content to a contribution region.
OK	Saves your settings and closes the Site Address dialog.
Cancel	Cancels your settings and closes the Site Address dialog.
Help	Opens the online help for this specific dialog.

Before a domain name address works, you must properly configure the domain name (DNS record) on a name server.

The default folder address you see when you first open this dialog is the one used by Oracle Site Studio when the site was first created.

A.6 Choose Default Link Format Dialog

When you create a hyperlink in Designer and in Contributor, you have numerous choices regarding the format of the link. Using this dialog, you can choose a default link format for designers and contributors, and you can hide this option so that it does not appear in the Link Wizard (see [Section A.18, "Link Wizard"](#)).

Element	Description
Path-based URL	Absolute Path: Generates a full path. For example: <!--\$ssServerRelativeSiteRoot-->products/index.htm (where "<!--\$ssServerRelativeSiteRoot-->" gets replaced with the path to the root of the web site). Relative path: Uses a relative path instead of a full path. For example: ../products/index.htm
ID-based URL	Contains the hard-coded ID of the target location without referencing a section name or label. Choose an option: Client Side javascript:link() / javascript:nodelink() format: Uses client-side JavaScript and the ID of the target location. Server Side wcmUrl() format: Uses server-side Idoc Script and the ID of the target location. URL Token ssLINK / ssNODELINK format: Uses an Oracle Site Studio token and the ID of the target location.
Hide the "Choose Format" page in the Designer Hyperlink Wizard	Prevents the Create Hyperlink Wizard - Choose Format screen (see Section A.18.3, "Link Wizard - Choose URL Format") from displaying in the Create Hyperlink Wizard for designers.
Hide the "Choose Format" page in the Contributor Hyperlink Wizard	Prevents the Create Hyperlink Wizard - Choose Format screen (see Section A.18.3, "Link Wizard - Choose URL Format") from displaying in the Create Hyperlink Wizard for contributors.
OK	Saves your settings and closes the Choose Default Link Format dialog.

Element	Description
Cancel	Cancels your settings and closes the Choose Default Link Format dialog.
Help	Opens the online Help.

A.7 Define Environment Properties Dialog

When replicating a web site, the properties of the site are replicated as well from one server to the next. In this dialog, you can specify properties that are not replicated and instead remain on the source server. The settings are called environment properties.

Element	Description
Name	The name of the property. Check the box beside the name to turn it into an environment property.
Type	The type of property.
Description	A description of the property.
OK	Saves your settings and closes the Environment Properties dialog.
Cancel	Cancels your settings and closes the Environment Properties dialog.
Help	Opens the online help for this specific dialog.

A.8 New Section Dialog

Each section in the site hierarchy contains a label, a URL directory name, and a section ID. These values are used repeatedly throughout your site. You can choose these values when you first create a section, and you can edit them after you create the section (using the Site Hierarchy pane and the Properties pane).

Element	Description
Label	The section label displays in the Site Hierarchy pane and in the navigation menu on your web site (generated by navigation fragments).
URL	The URL is used in the web browser address bar, and it appears in all (path-based) hyperlinks that target the section. This value should not contain spaces or special characters, as they can create linking problems on the web site. Oracle Site Studio automatically strips out illegal characters as you enter the label.
Auto-generate section ID	Automatically generates a section ID (also referred to as "nodeID") that is used in certain (ID-based) hyperlinks that target the section.
Manually enter section ID	Select to enter the section ID (also referred to as "nodeID") yourself. This value is used in certain (ID-based) hyperlinks that target the section.
OK	Saves your settings and closes the New Section dialog.
Cancel	Cancels your settings and closes the New Section dialog.
Help	Opens the online Help for this specific dialog.

A.9 Custom Section Properties Dialog

Each section in the site hierarchy contains properties (ID, label, default placeholder definition, and so on). In addition to these, you can create a custom property and then modify the web site to read that value and consequently, perform a certain function. The custom section properties that you add to your site are listed here. You can edit or delete them at any time using this dialog.

Element	Description
Name Type Description (Viewing area)	Displays existing custom section properties.
Add	Opens the Define Custom Section Property dialog (see Section A.10, "Define Custom Section Property Dialog"), where you can add a custom section property.
Delete	Deletes a custom section property.
Edit	Opens the Define Custom Section Property dialog (see Section A.10, "Define Custom Section Property Dialog"), where you can edit a custom section property.
OK	Saves your settings and closes the Define Custom Section Properties dialog.
Cancel	Cancels your settings and closes the Define Custom Section Properties dialog.
Help	Opens the online help for this specific dialog.

A.10 Define Custom Section Property Dialog

Each section in the site hierarchy contains properties (ID, label, default placeholder definition, and so on). In addition to these, you can create a custom property and then modify the web site to read that value and consequently, perform certain function. You can add and edit a custom section property using this dialog.

Element	Description
Name	The name of the custom section property. The name displays in the Properties pane when you select a section in the Site Hierarchy pane.

Element	Description
Type	<p>The type of action to take for this property (similar to the way a fragment parameter works).</p> <p>text: Provides a simple text box that you can use to add text.</p> <p>bigtext: Provides a much larger text box (in a popup window) where you can add and edit text.</p> <p>boolean: Provides a True or False value that you can choose from.</p> <p>integer: Provides an integer value. The integer value may not contain decimal points.</p> <p>float: Provides a value, including decimal points.</p> <p>size: Provides a sizing value, in pixels or percentage.</p> <p>color: Provides an RGB hexadecimal value (for example, 0xFF0000) and a color picker.</p> <p>url: Provides a dialog (see Section A.18.1, "Link Wizard - Choose Type of Link") that you use to browse to a URL and then select it as a value.</p> <p>manageddoc: Opens a content server window that you can use to browse to a document in the content server and then select it as a value (dDocName).</p> <p>managedurl: Opens a content server window that you can use to browse to the URL of a document in the content server and then select it as a value.</p> <p>managedquery: Opens the Edit Query Text dialog (see Section A.22, "Edit Query Text Dialog"), where you can build a query.</p> <p>cssstyle: Provides a text box where you can specify a CSS style value.</p> <p>siteid: Provides a dialog (see Section A.16, "Select Site Dialog") that you use to select a web site in the content server.</p> <p>nodeid: Opens a dialog (see Section A.17, "Select Section Dialog") that you use to select a section in your site hierarchy.</p>
Description	Optional setting used to display a description of the property at the bottom of the Properties pane.
OK	Adds the custom section property and closes the Define Custom Section Property dialog.
Cancel	Cancels your settings and closes the Define Custom Section Property dialog.
Help	Opens the online help for this specific dialog.

After you add the custom section property, you must go through your site hierarchy and enter a value for each section. Then, you must modify your site assets to read that custom property and perform a certain function.

A.11 Select (Home) Page Template Dialog

A home page is like a primary page, except that it is associated with the root of the web site rather than one of its sections. It is the default, or landing, page for visitors when they enter the site. You can add a home page using this dialog.

Element	Description
Create a new page template	<p>Creates the home page with the file extension that you specify below:</p> <p>hcsp: Use for Idoc Script web sites.</p> <p>jsp: Use for JSP web sites.</p> <p>asp: Use for ASP web sites.</p> <p>The available file extensions depend on the type of site you created using the Site Connection Details dialog (see Section A.3, "Site Connection Details Dialog").</p> <p>You can create a web site in HCSP, JSP, or ASP. But only those sites built in HCSP have the functionality that is in 10gR4 and later releases. You should not create a web site in ASP or JSP unless you have determined that you have very specific needs that require a site built in ASP or JSP.</p>
Select existing page template from the content server	<p>Adds an existing page template (from the content server) to your web site, making it the home page. You can select it from the menu if the page template is one you created recently; if not, select <Choose From Server> to browse to it in the content server when this dialog closes.</p>
Create a new page template from an existing one	<p>Creates a copy of an existing page template and adds it to your web site, making it the home page. This action saves you time by letting you reuse another page template rather than re-create the design from scratch.</p> <p>The page template that you are creating this from can come from one of two sources:</p> <p>stored in the content server: Selects a page template stored on the content server. You can select it from the menu if the page template is one you created recently; if not, select <Choose From Server> to browse to it in the content server when this dialog closes.</p> <p>stored on the file system: Selects a page template stored on the file system. This option can be especially useful if you created a page template on your file system and want to reuse it. The file, however, must have the appropriate file extension (.hcsp, .jsp, or .asp).</p>
Use an external URL	<p>This option applies only to primary pages (see Section A.12, "Select (Primary) Page Template Dialog").</p>
Apply recursively to all subsections	<p>Applies the same page template to every section below this one in the site hierarchy.</p> <p>This action is the same as opening each section in the site hierarchy, adding a page template, and choosing "Select existing page template from the content server."</p>
OK	<p>Closes the Select Page Template dialog and associates the page template with the root of your site hierarchy.</p>
Cancel	<p>Closes the Select Page Template dialog without associating a page template with the root of your site hierarchy.</p>
Help	<p>Opens the online help for this specific dialog.</p>

A.12 Select (Primary) Page Template Dialog

In most cases, each section in the site hierarchy contains a primary page, which acts as the default, or landing, page for that section. You can create a primary page for each section in the site hierarchy or reuse the same primary page throughout the site hierarchy. You can add a primary page using this dialog.

Element	Description
Create a new page template	<p>Creates a primary page for this section of the site, using the file extension that you specify below:</p> <p>hcsp: Use for Idoc Script web sites.</p> <p>jsp: Use for JSP web sites.</p> <p>asp: Use for ASP web sites.</p> <p>The available file extensions depend on the type of site you created (see Section A.3, "Site Connection Details Dialog").</p> <p>You can create a web site in HCSP, JSP, or ASP. But only those sites built in HCSP have the functionality that is in 10gR4 and later releases. You should not create a web site in ASP or JSP unless you have determined that you have very specific needs that require a site built in ASP or JSP.</p>
Select existing page template from the content server	<p>Adds an existing page template from the content server to your web site, making it the primary page for this section. You can select it from the menu if the page template is one you created recently; if not, select <Choose From Server> to browse to it in the content server when this dialog closes.</p>
Create a new page template from an existing one	<p>Creates a copy of an existing page template and adds it to your web site, making it the primary page for this section. This action saves you time by letting you reuse another page template rather than re-create the design from scratch.</p> <p>The page template used to create the new one may come from one of two sources:</p> <p>stored in the content server: A page template stored on the content server is used. You can select it from the menu if the page template is one you created recently; if not, select <Choose From Server> to browse to it in the content server when this dialog closes.</p> <p>stored on the file system: A page template stored on the file system is used. This option is especially useful if you created a page template on your file system and want to reuse it. This file must have the appropriate file extension (.hcsp, .jsp, or .asp).</p>
Use an external URL	<p>Uses the web page of another web site (possibly a partner site or department-level site) instead of a page template. When visitors go to this section, they are taken to the URL that you specify here.</p>
Apply recursively to all subsections	<p>Applies this same page template to every section below this one. This is the same as opening each section below this one in the site hierarchy, adding a page template, and choosing "Select existing page template from the content server."</p>
OK	<p>Closes the Select Page Template dialog and associates the primary page with this section in the site hierarchy.</p>
Cancel	<p>Closes the Select Page Template dialog without associating the primary page with this section in the site hierarchy.</p>
Help	<p>Opens the online help for this specific dialog.</p>

A.13 Select (Secondary) Page Template Dialog

In most instances, the secondary page is used for a list of items, or items found in a search. A secondary page can also serve as a backdrop for files (data files or native documents) added to the site by a contributor. As such, they are only required when you allow contributors to add files to the web site. You can add a secondary page to the root of your site and to a section of your site. Furthermore, you can create a secondary page or reuse a secondary page from another section. You add secondary pages using this dialog.

Element	Description
Create a new page template	<p>Creates a secondary page for this section of the site, using the file extension that you specify below:</p> <p>hcsp: Use for Idoc Script web sites.</p> <p>jsp: Use for JSP web sites.</p> <p>asp: Use for ASP web sites.</p> <p>The available file extensions depend on the type of site you created (see Section A.3, "Site Connection Details Dialog").</p> <p>You can create a web site in HCSP, JSP, or ASP. But only those sites built in HCSP have the functionality that is in 10gR4 and later releases. You should not create a web site in ASP or JSP unless you have determined that you have very specific needs that require a site built in ASP or JSP.</p>
Select existing page template from the content server	<p>Adds an existing page template from the content server to your web site, making it the secondary page for this section. You can select it from the menu if the page template is one you created recently; if not, select <Choose From Server> to browse to it in the content server when this dialog closes.</p>
Create a new page template from an existing one	<p>Creates a copy of an existing page template and adds it to your web site, making it the secondary page for this section. This action saves you time by letting you reuse another page template rather than re-create the design from scratch.</p> <p>The page template used to create the new one may come from one of two sources:</p> <p>stored in the content server: A page template stored on the content server is used. You can select it from the menu if the page template is one you created recently; if not, select <Choose From Server> to browse to it in the content server when this dialog closes.</p> <p>stored on the file system: A page template stored on the file system is used. This option is especially useful if you created a page template on your file system and want to reuse it. This file must have the appropriate file extension (.hcsp, .jsp, or .asp).</p>
Use an external URL	<p>This option applies only to primary pages (see Section A.12, "Select (Primary) Page Template Dialog").</p>
Apply recursively to all subsections	<p>Applies this same page template to every section below this one. This is the same as opening each section below this one in the site hierarchy, adding a page template, and choosing "Select existing page template from the content server."</p>
OK	<p>Closes the Select Page Template dialog and associates the secondary page with this section in the site hierarchy.</p>
Cancel	<p>Closes the Select Page Template dialog without associating the secondary page with this section in the site hierarchy.</p>
Help	<p>Opens the online help for this specific dialog.</p>

A.14 Site Asset Categories Dialog

You use the Site Asset Categories dialog to customize and even add categories to the Site Assets pane in Designer.

Element	Description
	<p>Creates a category. (A text box opens where you can enter your category and then click Enter.)</p>

Element	Description
	Deletes an existing category. (Existing files in that category are not deleted from the content server.)
	Moves the selected category up in the list.
	Moves the selected category down in the list.
Description	Describes the asset category.
Website Object Type	Indicates the Website Object Type for the asset.
Query Text	<p>The query that is performed in the content server to display the assets that belong to this category.</p> <p>Normally, the query picks up everything that contains the correct Type (from above), but you can use this value to further customize the assets that should appear in this category.</p> <p>The Additional Information button opens the Edit Query Text dialog (see Section A.22, "Edit Query Text Dialog"), where you specify, test, and capture a custom query.</p>
Metadata	Click to open the Enable Metadata Modification dialog (see Section A.15, "Enable Metadata Modification Dialog"), where you can make particular metadata available when you add an asset using the Site Assets pane.
OK	Saves your settings and closes the Site Asset Categories dialog.
Cancel	Cancels your settings and closes the Site Asset Categories dialog.
Help	Opens the online help for this specific dialog.
Reset to Default	Resets the default Site Asset categories (removing any categories that you may have created).

In order for a site asset to appear in the Site Assets pane, it must have the appropriate Websites and Website Object Type metadata assigned to it in the content server.

A.15 Enable Metadata Modification Dialog

You can specify the metadata that is available when you add an asset using the Site Assets pane. You can also specify the default metadata that is assigned to each new asset.

Element	Description
On Name Caption Type Default (Viewing area)	<p>Provides a checkbox for each metadata value in the content server.</p> <p>Check the box beside each metadata value to allow contributors to change that value when saving a contributor data file, or clear the checkbox to prevent the contributor from changing that metadata value.</p> <p>The Default column displays the metadata values that is assigned by default (see the Default Values button below).</p>
Check All	Checks all of the metadata values, making all of them available when a file is added using the Site Assets pane.
Clear All	Clears all of the metadata values so that they cannot be changed when a file is added using the Site Assets pane.
OK	Saves your settings and closes the Enable Metadata Modification dialog.
Cancel	Cancels your settings and closes the Enable Metadata Modification dialog.

Element	Description
Help	Opens the online help for this specific dialog.
Default Values	Opens the Assign Info Form (in the content server), where you specify the default metadata that is assigned to site assets added using the Site Assets pane. This is especially important if you disabled any of the metadata values in the Enable Metadata Modification dialog.
Clear Defaults	Clears the default metadata values.
Profile Trigger Value	Provides a box to enter the profile on the Oracle Content Server to honor the setup for that selected profile. This field appears only if profiles are enabled on the Oracle Content Server.

A.16 Select Site Dialog

You use this dialog when you must specify a web site in the content server where a piece of content should appear (also referred to as the target location). One such place is in a fragment parameter or custom section property of type "siteid."

Element	Description
Site Label Site ID Site Type	Displays the available web sites in the content server. Select the web site to use.
OK	Saves your settings and closes the Select Site dialog.
Cancel	Cancels your settings and closes the Select Site dialog.
Help	Opens the online help for this specific dialog.

A.17 Select Section Dialog

When identifying where a piece of content should appear (such as specifying the value for a fragment parameter of type "nodeid"), It may be necessary to select the section on your web site as the target location. You can do so using this dialog.

Element	Description
Site Hierarchy	Displays the available sections in the site hierarchy. Click the section to use it as a value.
Section ID	The node ID for a section. A value is inserted automatically if you select a section using the site hierarchy window. You can also manually enter a value.
OK	Saves your settings and closes the Select Section dialog.
Cancel	Cancels your settings and closes the Select Section dialog.
Help	Opens the online help for this specific dialog.

A.18 Link Wizard

The Link wizard can be used to create different types of links to different locations on your site or other sites.

- [Section A.18.1, "Link Wizard - Choose Type of Link"](#)
- [Section A.18.2, "Link Wizard - Choose a Section"](#)
- [Section A.18.3, "Link Wizard - Choose URL Format"](#)

- [Section A.18.4, "Link Wizard - Choose Content File"](#)
- [Section A.18.5, "Link Wizard - Choose Target Section"](#)
- [Section A.18.6, "Link Wizard - Check-In Content"](#)
- [Section A.18.7, "Link Wizard - Confirmation"](#)
- [Section A.18.8, "Link Wizard - Link Target Options"](#)

A.18.1 Link Wizard - Choose Type of Link

The first screen in the Link wizard asks you to choose the link type. Depending on your choice here, you see different options in the wizard.

Element	Description
Link to a section	Creates a link to another section on your site or a section on another site in the content server.
Link to a file	Creates a link to a contributor data file or native document on your site or another site in the content server. You have the option later on to create a link to a new or existing file.
Link to following URL	Creates a link to the URL that you specify in this text box.
Cancel	Cancels your settings and closes the Link wizard.
Back	Disabled on this screen.
Next	Takes you to the next screen in the wizard.
Help	Opens the online help for this specific dialog.

A.18.2 Link Wizard - Choose a Section

On this screen in the Link wizard, you identify the section to link to.

Element	Description
Menu	Lists the available sites in the content server that you can create a link to.
Site hierarchy window	Lists the sections on the selected site (from above) that you can create a link to.
Cancel	Cancels your settings and closes the Link wizard.
Back	Takes you to the previous screen in the wizard.
Next	Takes you to the next screen in the wizard.
Finish	Completes the wizard and inserts the link.
Help	Opens the online help for this specific dialog.

A.18.3 Link Wizard - Choose URL Format

In the Link wizard, you can use several different linking formats for the hyperlink. There are advantages and disadvantages to each format.

Element	Description
Path Based URL	<p>The link contains a path to the target location. You have two choices for this type of link:</p> <p>Absolute Path: Generates a full path. For example: <code><!--\$ssServerRelativeSiteRoot-->products/index.htm</code> (where "<code><!--\$ssServerRelativeSiteRoot--></code>" gets replaced with the path to the root of the web site).</p> <p>Relative Path: Generates a relative path instead of a full path. For example: <code>../products/index.htm</code>.</p>
ID Based URL	<p>The link contains the coded identity of the target location rather than the path-based name. You have three choices for this type of link.</p> <p>Client Side Script Format: Uses client-side JavaScript to construct a link to the target location. For example: <code>javascript:nodelink('123');</code>, <code>javascript:link('myfile');</code>, or <code>javascript:link('myfile','123');</code>.</p> <p>Server Side Script Format: Uses server-side Idoc Script to construct a link to the target location. For example: <code><!--\$ssNodeLink("123")--></code>, <code><!--\$ssLink("myfile")--></code>, or <code><!--\$ssLink("myfile","123")--></code>.</p> <p>URL Token Format: Uses a redirect on the server to construct a link to the target location. For example: <code>ssNODELINK/123</code>, <code>ssLINK/myfile</code>, or <code>ssLINK/123/myfile</code>.</p>
Cancel	Cancels your settings and closes the Link wizard.
Back	Takes you to the previous screen in the wizard.
Next	Takes you to the next screen in the wizard.
Finish	Completes the wizard and inserts the link.
Help	Opens the online help for this specific dialog.

If you know that you always want to choose the same format, you can hide this screen so that it does not display again, for both designers and contributors, using the Choose Default Link Format dialog (see [Section A.6, "Choose Default Link Format Dialog"](#)).

The "Path-based URL" option may be disabled, depending on what you chose on the previous screen (for example, if you're creating a link to a file and you choose its Web Site Section metadata as the target section).

A.18.4 Link Wizard - Choose Content File

In the Link wizard, you can create a link to a new or existing contributor data file or native document. If you create a link to a new file, you can choose the type of file from a list and Oracle Site Studio checks it in for you. If you choose an existing file, you can select one from the content server or browse for one locally.

Element	Description
New Contributor data file	Choose to check in a new contributor data file that is the target of the link.
New native file	Choose to check in a new native document to target in the link. Select the native document type in the menu.
Existing file from server	Choose to view the files associated with the current web site on the server. The file selected is the target for the link.
Existing local file	Choose to check in a local file to target in the link.

Element	Description
Current item	Choose to select the current file as the link target.
Cancel	Cancels your settings and closes the Link wizard.
Back	Takes you to the previous screen in the wizard.
Next	Takes you to the next screen in the wizard.
Help	Opens the online help for this specific dialog.

When you use the Link wizard to select a file or add a file to your site, Oracle Site Studio automatically assigns the appropriate metadata to the file so that it is recognized as part of your site.

A.18.5 Link Wizard - Choose Target Section

In the Link wizard, you can create a link to a contributor data file or native document, and you can control where that file appears on the web site when the link is clicked. This has no relation to where the file is actually stored on the web site or the content server. You can specify any location on your site or another site.

Alternatively, you can let Oracle Site Studio identify the section where the file is currently used in the web site.

Element	Description
Use default website section metadata	The file opens in the section where it is stored (using the metadata defined in xWebsiteSection). If you choose this value, you must use an ID-based link format on the next screen.
Choose a website section	The file opens in the section that you specify in the menu.
Menu	Lists the available sites in the content server. Select one of these sites as the target location for your file.
Site hierarchy window	Lists the available sections on the selected site (from above). Select one of these sections as the target location for your file.
Link to the Content Item's URL	The link created links directly to the item's web layout URL.
Options	Opens a link options dialog (see Section A.18.8, "Link Wizard - Link Target Options").
Cancel	Cancels your settings and closes the Link wizard.
Back	Takes you to the previous screen in the wizard.
Next	Takes you to the next screen in the wizard.
Finish	Completes the wizard and inserts the link.
Help	Opens the online help for this specific dialog.

A.18.6 Link Wizard - Check-In Content

When you create a link to a new file, or are linking to one hosted locally, then the item is checked into the content server. This section of the wizard is used to collect all of the necessary content server data. All fields marked in red are required; that is, the item cannot be checked in without providing values for these fields.

A.18.7 Link Wizard - Confirmation

On the last screen in the Link wizard, you see what your hyperlink looks like.

Element	Description
Link URL	Shows what the generated link looks like. You can view this link as is or copy and paste it into another file.
Options	Opens a link options dialog (see Section A.18.8, "Link Wizard - Link Target Options").
Cancel	Cancels your settings and closes the Link wizard.
Back	Takes you to the previous screen in the wizard.
Finish	Completes the wizard and inserts the link.
Help	Opens the online help for this specific dialog.

A.18.8 Link Wizard - Link Target Options

The options of the link are available in this smaller dialog from any point in the wizard. Here you can control whether the link opens in a new window or not.

Element	Description
Open link target in new window	When checked, the link opens in a new browser window. _blank : The value specified when the link is opened.
Ok	Accepts the entered options and returns you to the screen the dialog was launched from.
Cancel	Cancels your settings and closes the Link Target Options dialog.

Choosing the "Open link target in new window" is useful when you want to prevent visitors from inadvertently leaving the web page when they click the link.

A.19 Contribution Region Dialog

A contribution region allows contributors to edit a portion of the web page using the Contributor application or a third-party application (when native documents are used). You can add a contribution region to a page template, specify the options that are available to contributors, and customize document conversion settings (for native documents) in this dialog.

Element	Description
ID	An identification automatically generated by Oracle Site Studio.
Name	The name that you give to the contribution region. The name displays in Design view to represent the region. (The name of the region is used for coding purposes and may not contain spaces, or non-ASCII or special characters.) If you reuse the same data file on multiple page templates, then the name of the region and its elements on each page must be the same.
Contributor edit	Makes the contribution region editable. Consequently, contributors can edit the content assigned to the region by left-clicking the contribution icon or right-clicking the icon and choosing Edit (to launch Contributor for data files) or Check Out and Open (to launch a third party application for native documents).

Element	Description
Document info	Adds a "Document Info" link to the right-click menu of the contribution icon. Contributors can use this to go to the Content Information page for the data file or native document assigned to the region.
Workflow approve	Adds an "Approve Document" link to the right-click menu of the contribution icon. Contributors (contributors designated as reviewers or contributors in the workflow) use this to approve the data file or native document assigned to the region.
Workflow reject	Adds a "Reject Document" link to the right-click menu of the contribution icon. Reviewers (contributors designated as reviewers or contributors in the workflow) use this to reject the data file or native document assigned to the region.
Modify metadata	Takes contributors to the Assign Info Form page before saving the data file assigned to the region so that they can update the metadata for the data file. Define: Opens the Enable Metadata Modification dialog (see Section A.15, "Enable Metadata Modification Dialog"), where you can specify the metadata that is available to contributors when they save the file assigned to the region.
Switch region content	Adds a "Switch Region Content" link to the right-click menu of the contribution icon. Contributors use this to create a file (contributor data file or native document) or browse to an existing file and assign it to the region. Define: Opens the Region Content Options dialog (see Section A.20, "Region Content Options Dialog"), where you specify the options that are available to a contributor when switching or assigning content to the region.
Switch region template	Adds a "Switch Region Template" link to the right-click menu of the contribution icon. Contributors use this to select a different region template to use in the contribution region.
View Web Site Usage Report	Adds a "View Web Site Usage Report" link to the right-click menu of the contribution icon. Contributors use this to view a site report showing where the file (the contributor data file or native document assigned to the region) is used throughout the site.
Settings	Opens the Native Document Conversion Settings dialog (see Section A.24, "Native Document Conversion Settings Dialog (Legacy)"), where you specify a Dynamic Converter template or rule that is used to convert native documents assigned to the region.
OK	Saves your settings and closes the Contribution Region dialog.
Cancel	Cancels your settings and closes the Contribution Region dialog.
Help	Opens the online help for this specific dialog.

A.20 Region Content Options Dialog

The types of files that a contributor can assign are defined using this dialog.

Element	Description
Create new contributor data file	Allows a contributor to create a contributor data file and assign it.

Element	Description
Create new native document	<p>Allows a contributor to create a native document and assign it.</p> <p>Document Types: Specifies which native documents a contributor can create.</p> <p>The Additional Information button opens the Choose Document Types dialog (see Section A.21, "Choose Document Types Dialog"), which you can use to select native document types from a list.</p>
Choose existing in the content server	<p>Allows a contributor to search the content server for an existing contributor data file or native document and assign it.</p> <p>Query Text: Specifies the actual query that is performed in the content server to display existing files.</p> <p>The Additional Information button opens the Edit Query Text dialog (see Section A.22, "Edit Query Text Dialog"), which you can use to create, capture, and test the query used to display existing files.</p> <p>Show Results Only: Shows only the search results and not the content server environment. (This is useful when you want to prevent contributors from browsing to other content in the content server.)</p>
Browse for content locally	Allows a contributor to browse for a file on the local file system and check that file in.
Remove region content association	Allows the contributor to remove an existing contributor data file or native document already assigned. (This adds a "None" option to the Choose Region Content dialog.)
Default Metadata	Opens the Enable Metadata Modification dialog (see Section A.15, "Enable Metadata Modification Dialog"), where you specify the metadata that is available to the contributor and the default metadata that is assigned when a contributor creates a file.
OK	Saves your settings and closes the Region Content Options dialog.
Cancel	Cancels your settings and closes the Region Content Options dialog.
Help	Opens the online help for this specific dialog.

A.21 Choose Document Types Dialog

When you allow a contributor to create a native document and add it to the site, you must specify the type of native document that they can create (for example, Microsoft Word, Excel, and PowerPoint).

Element	Description
List of file types	<p>Displays the native documents that are available on your system. Check the box beside the type of document to allow contributors to create.</p> <p>Not all document types in this dialog are supported for contribution (for example, image authoring software).</p>
OK	Saves your settings and closes the Choose Document Types dialog.
Cancel	Cancels your settings and closes the Choose Document Types dialog.
Help	Opens the online help for this specific dialog.

A.22 Edit Query Text Dialog

There are numerous occasions when you want to specify a custom query to perform in the content server. This query searches for files matching one or more metadata values

and then opens those results in Designer, Contributor, or the web site (depending on the element you're using).

You can specify, test, and capture your query using this dialog.

Element	Description
[Text field]	Use this text box to specify your query.
Capture Query	Opens the Content Query Capture screen in the content server that you use to perform a query and then save (capture) that query.
Test Query	Use to test your query in the content server.
OK	Saves your settings and closes the Edit Query Text dialog.
Cancel	Cancels your settings and closes the Edit Query Text dialog.

A.23 Conversion Definition Dialog

The conversion definition dialog is used to organize and assign the conversion template or rule as a site asset. After a conversion definition is created and named, then it can be used as a site asset.

Element	Description
Conversion Definition	Each named conversion definition is listed.
Add	Opens the Native Document Conversion Settings dialog (see Section A.25, "Native Document Conversion Settings Dialog") to add a dynamic conversion template or rule to a conversion definition.
Remove	Removes the selected conversion definition.
Edit	Opens the data for the selected definition in the Native Document Conversion Settings dialog (see Section A.25, "Native Document Conversion Settings Dialog") to edit the selected conversion definition.
Help	Opens the online help for this specific dialog.

A.24 Native Document Conversion Settings Dialog (Legacy)

When you assign a native document to a contribution region, you can specify a (Dynamic Converter) template or rule that is used to convert the document into a web page. You do so using this dialog.

Element	Description
Use simple inline dynamic conversion	A blank template is used to convert the native document. This template contains minimal conversion settings, and therefore the resulting web page appears similar to the original document (which may or may not be desirable).

Element	Description
Use full dynamic conversion	<p>Specifies a Dynamic Converter conversion template and layout page template currently checked into the content server.</p> <p>Choose Template: Opens a content server window where you can select a conversion template.</p> <p>Choose Layout: Opens a content server window where you can select a Dynamic Converter layout page.</p> <p>For the layout, we recommend using "snippet_layout.txt" because it strips out the opening and closing HTML tags. This allows the converted document to be inserted into another web page, which is necessary for Oracle Site Studio.</p>
Use single rule dynamic conversion	Specifies a single conversion rule that evaluates and converts the native document. You can select the rule from the menu.
Use rules engine dynamic conversion	Specifies that the default rule evaluation be applied to evaluate and convert the native document.
Use custom dynamic conversion command	<p>Specifies a custom script that is used to convert the native document.</p> <p>You can use this option to add your own Idoc Script to the conversion sequence. If you choose this, you can copy the existing conversion syntax from the Command string area (below), paste it into this field, and add your custom script to it.</p> <p>A dynamic conversion does not take place if the conversion command is missing or invalid.</p>
Command string	Displays the conversion syntax that is used to convert the native document.
OK	Saves your settings and closes the Native Document Conversion Settings dialog.
Cancel	Cancels your settings and closes the Native Document Conversion Settings dialog.
Help	Opens the online help for this specific dialog.

Before using this feature, you must create a conversion template or rule in Dynamic Converter.

A.25 Native Document Conversion Settings Dialog

When you add a native document to a web page, the conversion definitions can specify a (Dynamic Converter) template or rule that is used to convert the document into a web page. You do so using this dialog.

Element	Description
Name	The name of the conversion for identification.
Use simple inline dynamic conversion	A blank template is used to convert the native document. This template contains minimal conversion settings, and therefore the resulting web page appears similar to the original document (which may or may not be desirable).

Element	Description
Use full dynamic conversion	<p>Specifies a Dynamic Converter conversion template and layout page currently checked into the content server.</p> <p>Choose Template: Opens a content server window where you can select a conversion template.</p> <p>Choose Layout: Opens a content server window where you can select a Dynamic Converter layout page.</p> <p>For the layout, we recommend using "snippet_layout.txt" because it strips out the opening and closing HTML tags. This allows the converted document to be inserted into another web page, which is necessary for Oracle Site Studio.</p>
Use single rule dynamic conversion	Specifies a single conversion rule that evaluates and converts the native document. You can select the rule from the menu.
Use rules engine dynamic conversion	Specifies that the default rule evaluation be applied to evaluate and convert the native document.
Use custom dynamic conversion command	<p>Specifies a custom script that is used to convert the native document.</p> <p>You can use this option to add your own Idoc Script to the conversion sequence. If you choose this, you can copy the existing conversion syntax from the Command string area (below), paste it into this field, and add your custom script to it.</p> <p>A dynamic conversion does not take place if the conversion command is missing or invalid.</p>
Command string	Displays the conversion syntax that is used to convert the native document.
OK	Saves your settings and closes the Native Document Conversion Settings dialog.
Cancel	Cancels your settings and closes the Native Document Conversion Settings dialog.
Help	Opens the online help for this specific dialog.

A.26 WYSIWYG Element Dialog

The WYSIWYG element becomes a field in Contributor where users (contributors) add, edit, and delete text, graphics, and more. You can control the contributor's experience using this dialog. This particular element is used in legacy sites.

Element	Description
ID	The identification that Oracle Site Studio assigns to the element.
Name	<p>The name you assign to the element. It should not contain spaces, or non-ASCII or special characters.</p> <p>If you reuse the same data file in multiple places, then the name of the region and its elements on each page must be the same.</p>
Label	Briefly describes the element. The label displays beside the field in Contributor to let individual contributors know what content should go there.
Info	Provides a more detailed description of the element. This description appears as a tooltip when contributors hover their mouse over the Info icon in Contributor.
Type	Indicates the type of element you are adding.

Element	Description
Actions	<p>Actions determine the available editing options for a contributor:</p> <p>Undo: Contributors can undo their last edit.</p> <p>Redo: Contributors can redo their last edit.</p> <p>Cut: Contributors can cut text and images.</p> <p>Copy: Contributors can copy text and images.</p> <p>Paste: Contributors can paste text and images.</p> <ul style="list-style-type: none"> ▪ Formatted (on/off): Allows formatting or removes formatting when text is pasted in Contributor. <p>Insert Link: Contributors can create hyperlinks.</p> <ul style="list-style-type: none"> ▪ Link Settings: Opens the Link Settings dialog (see Section A.44, "Link Settings Dialog"), where you specify a contributor's linking options. <p>Remove Link: Contributors can remove hyperlinks.</p> <p>Edit hyperlink targets: Contributors can edit the file that is the target of a hyperlink (using the Element menu and right-click menu).</p> <p>Bold: Contributors can make the text bold.</p> <p>Italic: Contributors can make the text italic.</p> <p>Underline: Contributors can underline the text.</p> <p>Remove Formatting: Contributors can remove text formatting.</p> <p>Ordered List: Contributors can add an ordered (numbered) list.</p> <p>Unordered List: Contributors can add an unordered (bulleted) list.</p> <p>Indent: Contributors can indent text and images.</p> <p>Outdent: Contributors can outdent text and images.</p> <p>Left Justify: Contributors can left-align text and images.</p> <p>Center Justify: Contributors can center text and images.</p> <p>Right Justify: Contributors can right-align text and images.</p> <p>Insert Image: Contributors can add/replace images (gif, jpg, and png).</p> <ul style="list-style-type: none"> ▪ Set QueryText: Opens the Edit Query Text dialog (see Section A.22, "Edit Query Text Dialog"), where you specify the query used to display images to the contributor. ▪ Show Results Only (On/Off): Shows or hides the content server environment when a contributor searches for images. <p>Insert Horizontal Rule: Contributors can add a horizontal rule.</p> <p>Insert Line Break: Contributors can add a line break.</p> <p>Insert Non-Breaking Space: Contributors can add a non-breaking space.</p>

Element	Description
Actions (continued)	<p>Change Font Face: Contributors can change the font face.</p> <p>Change Font Size: Contributors can change the font size.</p> <p>Change Foreground Color: Contributors can change the text color.</p> <p>Change Background Color: Contributors can change the text background color.</p> <p>Spell Checker: Contributors can perform a spell check.</p> <p>Required: Requires information in the field before Contributor closes.</p> <p>Apply CSS Classes: Contributors can use different CSS classes.</p> <ul style="list-style-type: none"> ▪ Set Class Names: Opens the Select CSS Classes dialog (see Section A.45, "Select CSS Classes Dialog"), where you specify the classes that are available in Contributor. <p>Table Support: Contributors can add and edit tables.</p> <p>Validation: Used to validate the content added in Contributor (allowing you to enforce certain content requirements for contributors).</p> <ul style="list-style-type: none"> ▪ Validation Settings: Opens the WYSIWYG Element Validation dialog (see Section A.48, "WYSIWYG Element Validation Dialog"), where you specify what is validated. <p>Edit Object Properties: When selected, contributors can modify the object properties.</p> <p>Custom Configuration:</p> <ul style="list-style-type: none"> ▪ Select Custom Configuration: Opens the Custom Configuration Scripts dialog (see Section A.79, "Custom Configuration Scripts Dialog"). The number of configuration scripts selected appears to the right. <p>Accessibility Report: When selected, contributors can run an accessibility report on webpages from the WYSIWYG toolbar to see if the pages are compliant with multiple accessibility standards. Please note that this option is meaningful only if Ephox is used as the Contributor editor (which is not the default).</p> <p>Source Mode: Used to allow contributors to select to edit in WYSIWYG or HTML. If unselected, contributors can edit only in WYSIWYG.</p> <p>Expand Editor: Used to allow contributors to view the editing area full-screen. Please note that this option is meaningful only if Ephox is used as the Contributor editor (which is not the default).</p> <p>Apply HTML Tags: Used to determine which tags are available to Contributors who edit in HTML.</p> <ul style="list-style-type: none"> ▪ Set Tag Names: Open to select specific tags that are allowed for Contributors. For more information, see Section A.50, "Select HTML Tags Dialog." <p>Set Element Height: Used to set a fixed height of the element.</p> <ul style="list-style-type: none"> ▪ Set Height: Use to set the height of the element by entering a number and the unit of measurement (centimeters, millimeters, inches, percent, picas, pixels, points). <p>Do Not Enclose Text in Editor: Used to define how carriage returns are handled in HTML. If text is enclosed, each time the contributor enters a return, a paragraph is created. If not enclosed, a line break is inserted for each return. Please note that this option is meaningful only if Ephox is used as the Contributor editor (which is not the default).</p> <p>Override Editor CSS: Used to implement a style sheet for the editor window.</p>
Check All	Checks all of the actions so that a contributor has all available editing options.
Clear All	Clears all of the metadata values so that a contributor has no editing options.

Element	Description
OK	Saves your settings and closes the WYSIWYG Element dialog.
Cancel	Cancels your settings and closes the WYSIWYG Element dialog.
Help	Opens the online help for this specific dialog.

A.27 WYSIWYG Element Definition Dialog

The WYSIWYG element becomes a field in Contributor where users (contributors) add, edit, and delete text, graphics, and more. You can control the contributor's experience using this dialog.

Element	Description
Info	Provides a more detailed description of the element. This description appears as a tooltip when contributors hover their mouse over the Info icon in Contributor.
Type	Indicates the type of element you are adding.
Actions	<p>Actions determine the available editing options for a contributor:</p> <p>Undo: Contributors can undo their last edit.</p> <p>Redo: Contributors can redo their last edit.</p> <p>Cut: Contributors can cut text and images.</p> <p>Copy: Contributors can copy text and images.</p> <p>Paste: Contributors can paste text and images.</p> <ul style="list-style-type: none"> ▪ Formatted (on/off): Allows formatting or removes formatting when text is pasted in Contributor. <p>Insert Link: Contributors can create hyperlinks.</p> <ul style="list-style-type: none"> ▪ Link Settings: Opens the Link Settings dialog (see Section A.44, "Link Settings Dialog"), where you specify a contributor's linking options.

Element	Description
Actions (continued)	<p>Remove Link: Contributors can remove hyperlinks.</p> <p>Edit hyperlink targets: Contributors can edit the file that is the target of a hyperlink (using the Element menu and right-click menu).</p> <p>Insert bookmark: Contributors can insert a bookmarking link.</p> <p>Bold: Contributors can make the text bold.</p> <p>Italic: Contributors can make the text italic.</p> <p>Underline: Contributors can underline the text.</p> <p>Superscript: Contributors can make the text superscript.</p> <p>Subscript: Contributors can make the text subscript.</p> <p>Strikethrough: Contributors can mark the text with a strikethrough bar.</p> <p>Remove Formatting: Contributors can remove text formatting.</p> <p>Ordered List: Contributors can add an ordered (numbered) list.</p> <p>Unordered List: Contributors can add an unordered (bulleted) list.</p> <p>Indent: Contributors can indent (move to the right) text and images.</p> <p>Outdent: Contributors can outdent (move to the left) text and images.</p> <p>Left Justify: Contributors can left-align text and images.</p> <p>Center Justify: Contributors can center text and images.</p> <p>Right Justify: Contributors can right-align text and images.</p> <p>Full Justify: Contributors can fully justify the text and images.</p> <p>Blockquote: Contributors can place and align the text within a blockquote.</p> <p>Insert Image: Contributors can add/replace images (gif, jpg, and png).</p> <ul style="list-style-type: none"> ■ Set QueryText: Opens the Edit Query Text dialog (see Section A.22, "Edit Query Text Dialog"), where you specify the query used to display images to the contributor. ■ Show Results Only (On/Off): Shows or hides the content server environment when a contributor searches for images. <p>Insert Horizontal Rule: Contributors can add a horizontal rule.</p> <p>Insert Line Break: Contributors can add a line break.</p> <p>Insert Symbol: Contributors can add a special character from the symbol map.</p> <p>Edit hyperlink targets: Contributors can edit the file that is the target of a hyperlink (using the Element menu and right-click menu).</p> <p>Change Font Face: Contributors can change the font face.</p> <p>Change Font Size: Contributors can change the font size.</p> <p>Change Foreground Color: Contributors can change the text color.</p> <p>Change Background Color: Contributors can change the text background color.</p> <p>Spell Checker: Contributors can perform a spell check.</p> <p>Required: Requires information in the field before Contributor closes.</p> <p>Apply CSS Classes: Contributors can use different CSS classes.</p> <ul style="list-style-type: none"> ■ Set Class Names: Opens the Select CSS Classes dialog (see Section A.45, "Select CSS Classes Dialog"), where you specify the classes that are available in Contributor. <p>Table Support: Contributors can add and edit tables.</p> <p>Form support: Contributors can add and edit forms.</p>

Element	Description
Actions (continued)	<p>Validation: Used to validate the content added in Contributor (allowing you to enforce certain content requirements for contributors).</p> <ul style="list-style-type: none"> ▪ Validation Settings: Opens the WYSIWYG Element Validation dialog (see Section A.48, "WYSIWYG Element Validation Dialog"), where you specify what is validated. <p>Edit Object Properties: When selected, contributors can modify the object properties.</p> <p>Insert Flash: Contributors can insert Flash files.</p> <ul style="list-style-type: none"> ▪ Set QueryText: Opens the Edit Query Text dialog (see Section A.22, "Edit Query Text Dialog"), where you specify the query used to display images to the contributor. ▪ Show Results Only (On/Off): Shows or hides the content server environment when a contributor searches for images. <p>Custom Configuration: Used to allow configuration scripts to modify the contributor's editing window.</p> <ul style="list-style-type: none"> ▪ Select Custom Configuration: Opens the Custom Configuration Scripts dialog (see Section A.79, "Custom Configuration Scripts Dialog"). The number of configuration scripts selected appears to the right. <p>Accessibility Report: When selected, contributors can run an accessibility report on webpages from the WYSIWYG toolbar to see if the pages are compliant with multiple accessibility standards. Please note that this option is meaningful only if Ephox is used as the Contributor editor (which is not the default).</p> <p>Select All: Contributors can select everything within the editor.</p> <p>Print: Contributors can print the content as it appears in the editor. Content is not formatted as it would appear on the web page.</p> <p>Find and Replace: Contributors can perform find and replace functions when editing.</p> <p>Show HTML Elements: Contributors can view where HTML tags are used with the data and how they are used.</p> <p>Source Mode: Used to allow contributors to select to edit in WYSIWYG or HTML. If unselected, contributors can edit only in WYSIWYG.</p> <p>Expand Editor: Used to allow contributors to view the editing area full-screen.</p> <p>Apply HTML Tags: Used to determine which tags are available to Contributors who edit in HTML.</p> <ul style="list-style-type: none"> ▪ Set Tag Names: Open to select specific tags that are allowed for Contributors. For more information, see Section A.50, "Select HTML Tags Dialog." <p>Set Element Height: Used to set a fixed height of the element.</p> <ul style="list-style-type: none"> ▪ Set Height: Use to set the height of the element by entering a number and the unit of measurement (centimeters, millimeters, inches, percent, picas, pixels, points). <p>Do Not Enclose Text in Editor: Used to define how carriage returns are handled in HTML. If text is enclosed, each time the contributor enters a return, a paragraph is created. If not enclosed, a line break is inserted for each return.</p> <p>Override Editor CSS: Used to implement a style sheet for the editor window.</p> <ul style="list-style-type: none"> ▪ Select CSS File: Displays the CSS files on the content available to use.
Check All	Checks all of the actions so that a contributor has all available editing options.
Clear All	Clears all of the metadata values so that a contributor has no editing options.

Element	Description
OK	Saves your settings and closes the WYSIWYG Element dialog.
Cancel	Cancels your settings and closes the WYSIWYG Element dialog.
Help	Opens the online help for this specific dialog.

A.28 Plain Text Element Dialog

The Plain Text element becomes a field in Contributor where users perform basic (and somewhat limited) edits, such as cut, copy, and paste. This can be useful when you want to prevent contributors from formatting text (and thereby creating more consistent, or possibly cleaner web pages). This particular element is used in legacy sites.

Element	Description
ID	The identification that Oracle Site Studio assigns to the element.
Name	The name you assign to the element. This is used for coding purposes and as such should not contain spaces, or non-ASCII or special characters. If you reuse the same data file in multiple places, then the name of the region and its elements on each page must be the same.
Label	Briefly describes the element. The label opens beside the field in Contributor to let individual contributors know what content should go there.
Info	Provides a more detailed description of the element. This description appears as a tooltip when contributors hover their mouse over the Info icon in Contributor.
Type	Indicates the type of element you are adding.
Actions	<p>Actions determine the available editing options for a contributor:</p> <p>Undo: Contributors can undo their last edit.</p> <p>Redo: Contributors can redo their last edit.</p> <p>Cut: Contributors can cut text.</p> <p>Copy: Contributors can copy text.</p> <p>Paste: Contributors can paste text.</p> <ul style="list-style-type: none"> Formatted (on/off): Allows formatting or removes formatting when text is pasted in Contributor (For a Plain Text element, this means text includes HTML formatting tags or no formatting tags). <p>Spell Checker: Contributors can spell check their work.</p> <p>Required: Requires contributors to enter information in the field before Contributor closes.</p> <p>Validation: Used to validate the content added in Contributor (allowing you to enforce certain content requirements for contributors).</p> <ul style="list-style-type: none"> Validation Settings: Opens the Plain Text Element Validation dialog (see Section A.49, "Plain Text Element Validation Dialog"), where you specify what is validated. <p>Custom Configuration:</p> <ul style="list-style-type: none"> Select Custom Configuration: Shows or hides the content server environment when a contributor selects a custom configuration. <p>Set Element Height: Used to set a fixed height of the element.</p> <ul style="list-style-type: none"> Set Height: Use to set the height of the element by entering a number and the unit of measurement (centimeters, millimeters, inches, percent, picas, pixels, points).

Element	Description
Check All	Checks all of the actions so that a contributor has all available editing options.
Clear All	Clears all of the metadata values so that a contributor has no editing options.
OK	Saves your settings and closes the Plain Text Element dialog.
Cancel	Cancels your settings and closes the Plain Text Element dialog.
Help	Opens the online help for this specific dialog.

A.29 Plain Text Element Definition Dialog

The Plain Text element becomes a field in Contributor where users perform basic (and somewhat limited) edits, such as cut, copy, and paste. This can be useful when you want to prevent contributors from formatting text (and thereby creating more consistent, or possibly cleaner web pages).

Element	Description
Info	Provides a more detailed description of the element. This description appears as a tooltip when contributors hover their mouse over the Info icon in Contributor.
Type	Indicates the type of element you are adding.
Actions	<p>Actions determine the available editing options for a contributor:</p> <p>Undo: Contributors can undo their last edit.</p> <p>Redo: Contributors can redo their last edit.</p> <p>Cut: Contributors can cut text.</p> <p>Copy: Contributors can copy text.</p> <p>Paste: Contributors can paste text.</p> <p>Insert Symbol: Contributors can add a special character from the symbol map.</p> <p>Spell Checker: Contributors can spell check their work.</p> <p>Required: Requires contributors to enter information in the field before Contributor closes.</p> <p>Validation: Used to validate the content added in Contributor (allowing you to enforce certain content requirements for contributors).</p> <ul style="list-style-type: none"> ▪ Validation Settings: Opens the Plain Text Element Validation dialog (see Section A.49, "Plain Text Element Validation Dialog"), where you specify what is validated. <p>Custom Configuration: Used to allow configuration scripts to modify the contributor's editing window.</p> <ul style="list-style-type: none"> ▪ Select Custom Configuration: Opens the Custom Configuration Scripts dialog (see Section A.79, "Custom Configuration Scripts Dialog"). The number of configuration scripts selected appears to the right. <p>Select All: Contributors can select everything within the editor.</p> <p>Print: Contributors can print the content as it appears in the editor. Content is not formatted as it would appear on the web page.</p> <p>Find and Replace: Contributors can perform find and replace functions when editing.</p>

Element	Description
Actions	Set Element Height: Used to set a fixed height of the element.
(continued)	<ul style="list-style-type: none"> ▪ Set Height: Use to set the height of the element by entering a number and the unit of measurement (centimeters, millimeters, inches, percent, picas, pixels, points).
	Override Editor CSS: Used to implement a style sheet for the editor window.
	<ul style="list-style-type: none"> ▪ Select CSS File: Displays the CSS files on the content available to use.
Check All	Checks all of the actions so that a contributor has all available editing options.
Clear All	Clears all of the metadata values so that a contributor has no editing options.
OK	Saves your settings and closes the Plain Text Element dialog.
Cancel	Cancels your settings and closes the Plain Text Element dialog.
Help	Opens the online help for this specific dialog.

A.30 Image Element Dialog

The Image element becomes a field in Contributor where users can perform basic edits that relate to handling images, such as inserting an image, inserting a link, and applying a CSS class. This can be useful when you want to limit a contributor's capabilities to simply editing images. This particular element is used in legacy sites.

Element	Description
ID	The identification that Oracle Site Studio assigns to the element.
Name	<p>The name you assign to the element. This is used for coding purposes and as such should not contain spaces, or non-ASCII or special characters.</p> <p>If you reuse the same data file in multiple places, then the name of the region and its elements on each page must be the same.</p>
Label	Briefly describes the element. The label opens beside the field in Contributor to let individual contributors know what content should go there.
Info	Provides a more detailed description of the element. This description appears as a tooltip when contributors hover their mouse over the Info icon in Contributor.
Type	Indicates the type of element you are adding.

Element	Description
Actions	<p>Actions determine the available editing options for a contributor:</p> <p>Undo: Contributors can undo their last edit.</p> <p>Redo: Contributors can redo their last edit.</p> <p>Required: Requires information in the field before Contributor closes.</p> <p>Insert Link: Contributors can create hyperlinks.</p> <ul style="list-style-type: none"> ▪ Link Settings: Opens the Link Settings dialog (see Section A.44, "Link Settings Dialog"), where you specify a contributor's linking options. <p>Remove Link: Contributors can remove hyperlinks.</p> <p>Edit hyperlink targets: Contributors can edit the file that is the target of a hyperlink (using the Element menu and right-click menu).</p> <p>Apply CSS Classes: Contributors can use different CSS classes.</p> <ul style="list-style-type: none"> ▪ Set Class Names: Opens the Select CSS Classes dialog (see Section A.45, "Select CSS Classes Dialog"), where you specify the classes that are available in Contributor. <p>Insert Image: Contributors can add/replace images (gif, jpg, and png).</p> <ul style="list-style-type: none"> ▪ Set QueryText: Opens the Edit Query Text dialog (see Section A.22, "Edit Query Text Dialog"), where you specify the query that is used in the content server to display images to the contributor. ▪ Show Results Only (On/Off): Shows or hides the content server environment when a contributor searches for images. <p>Validation: Used to validate the content added in Contributor (allowing you to enforce certain content requirements for contributors).</p> <ul style="list-style-type: none"> ▪ Validation Settings: Opens the Image Element Validation dialog (see Section A.51, "Image Element Validation Dialog"), where you specify what is validated. <p>Edit Object Properties: When selected, contributors can modify the object properties.</p> <p>Custom Configuration:</p> <ul style="list-style-type: none"> ▪ Select Custom Configuration: Opens the Custom Configuration Scripts dialog (see Section A.79, "Custom Configuration Scripts Dialog"). The number of configuration scripts selected appears to the right. <p>Set Element Height: Used to set a fixed height of the element.</p> <ul style="list-style-type: none"> ▪ Set Height: Use to set the height of the element by entering a number and the unit of measurement (centimeters, millimeters, inches, percent, picas, pixels, points).
Check All	Checks all of the actions so that a contributor has all available editing options.
Clear All	Clears all of the metadata values so that a contributor has no editing options.
OK	Saves your settings and closes the Image Element dialog.
Cancel	Cancels your settings and closes the Image Element dialog.
Help	Opens the online help for this specific dialog.

A.31 Image Element Definition Dialog

The Image element becomes a field in Contributor where users can perform basic edits that relate to handling images, such as inserting an image, inserting a link, and applying a CSS class. This can be useful when you want to limit a contributor's capabilities to simply editing images.

Element	Description
Info	Provides a more detailed description of the element. This description appears as a tooltip when contributors hover their mouse over the Info icon in Contributor.
Type	Indicates the type of element you are adding.
Actions	<p>Actions determine the available editing options for a contributor:</p> <p>Undo: Contributors can undo their last edit.</p> <p>Redo: Contributors can redo their last edit.</p> <p>Required: Requires information in the field before Contributor closes.</p> <p>Insert Link: Contributors can create hyperlinks.</p> <ul style="list-style-type: none"> ▪ Link Settings: Opens the Link Settings dialog (see Section A.44, "Link Settings Dialog"), where you specify a contributor's linking options. <p>Remove Link: Contributors can remove hyperlinks.</p> <p>Edit hyperlink targets: Contributors can edit the file that is the target of a hyperlink (using the Element menu and right-click menu).</p>
Actions (continued)	<p>Apply CSS Classes: Contributors can use different CSS classes.</p> <ul style="list-style-type: none"> ▪ Set Class Names: Opens the Select CSS Classes dialog (see Section A.45, "Select CSS Classes Dialog"), where you specify the classes that are available in Contributor. <p>Insert Image: Contributors can add/replace images (gif, jpg, and png).</p> <ul style="list-style-type: none"> ▪ Set QueryText: Opens the Edit Query Text dialog (see Section A.22, "Edit Query Text Dialog"), where you specify the query that is used in the content server to display images to the contributor. ▪ Show Results Only (On/Off): Shows or hides the content server environment when a contributor searches for images. <p>Validation: Used to validate the content added in Contributor (allowing you to enforce certain content requirements for contributors).</p> <ul style="list-style-type: none"> ▪ Validation Settings: Opens the Image Element Validation dialog (see Section A.51, "Image Element Validation Dialog"), where you specify what is validated. <p>Edit Object Properties: When selected, contributors can modify the object properties.</p> <p>Custom Configuration: Used to allow configuration scripts to modify the contributor's editing window.</p> <ul style="list-style-type: none"> ▪ Select Custom Configuration: Opens the Custom Configuration Scripts dialog (see Section A.79, "Custom Configuration Scripts Dialog"). The number of configuration scripts selected appears to the right. <p>Select All: Contributors can select everything within the editor.</p> <p>Print: Contributors can print the content as it appears in the editor. Content is not formatted as it would appear on the web page.</p> <p>Set Element Height: Used to set a fixed height of the element.</p> <ul style="list-style-type: none"> ▪ Set Height: Use to set the height of the element by entering a number and the unit of measurement (centimeters, millimeters, inches, percent, picas, pixels, points). <p>Override Editor CSS: Used to implement a style sheet for the editor window.</p> <ul style="list-style-type: none"> ▪ Select CSS File: Displays the CSS files on the content available to use.
Check All	Checks all of the actions so that a contributor has all available editing options.
Clear All	Clears all of the metadata values so that a contributor has no editing options.
OK	Saves your settings and closes the Image Element dialog.

Element	Description
Cancel	Cancels your settings and closes the Image Element dialog.
Help	Opens the online help for this specific dialog.

A.32 Custom Element Dialog

The Custom element becomes a field in Contributor that users (contributors) can use to add content that cannot be added using the other elements (WYSIWYG, Plain Text, and Image). More specifically, the Custom element provides a way to create a custom interface (also called a custom element form) that is used to add various types of content (multimedia files, source code, and so on). This particular element is used in legacy sites.

Element	Description
ID	The identification that Oracle Site Studio assigns to the element.
Name	The name you assign to the element. This name is used for coding purposes and as such should not contain spaces, or non-ASCII or special characters. If you reuse the same data file in multiple layout places, then the name of the region and its elements on each page must be the same.
Label	Briefly describes the element. The label opens beside the field in Contributor to let individual contributors know what content should go there.
Info	More detailed description of the element. This description appears as a tooltip when contributors hover their mouse over the Info icon in Contributor.
Type	Indicates the type of element you are adding.
Settings	Opens the Custom Element Settings dialog (see Section A.34, "Custom Element Settings Dialog"), where you can specify the Custom element form that is used by contributors.
Actions	Set Element Height: Used to set a fixed height of the element. <ul style="list-style-type: none"> ▪ Set Height: Use to set the height of the element by entering a number and the unit of measurement (centimeters, millimeters, inches, percent, picas, pixels, points).
OK	Saves your settings and closes the Custom Element dialog.
Cancel	Cancels your settings and closes the Custom Element dialog.
Help	Opens the online help for this specific dialog.

A.33 Custom Element Definition Dialog

The Custom element becomes a field in Contributor that users (contributors) can use to add content that cannot be added using the other elements (WYSIWYG, Plain Text, and Image). More specifically, the Custom element provides a way to create a custom interface (also called a custom element form) that is used to add various types of content. Custom elements can also be defined to help data-driven web sites publish based on values, as the custom elements can define boolean value, integer values, and other methods of sorting and filtering data for placement on the web page.

Element	Description
Info	More detailed description of the element. This description appears as a tooltip when contributors hover their mouse over the Info icon in Contributor.

Element	Description
Type	Indicates the type of element you are adding.
Settings	Opens the Custom Element Settings dialog (see Section A.34, "Custom Element Settings Dialog"), where you can specify the Custom element form that is used by contributors.
Actions	<p>Set Element Height: Used to set a fixed height of the element.</p> <ul style="list-style-type: none"> ▪ Set Height: Use to set the height of the element by entering a number and the unit of measurement (centimeters, millimeters, inches, percent, picas, pixels, points).
OK	Saves your settings and closes the Custom Element dialog.
Cancel	Cancels your settings and closes the Custom Element dialog.
Help	Opens the online help for this specific dialog.

A.34 Custom Element Settings Dialog

The Custom element becomes a field in Contributor that users (contributors) can use to add content that cannot be added using the other elements (WYSIWYG, Plain Text, and Image). More specifically, the Custom element provides a way to create a custom interface (also called a custom element form) that is used to add various types of content (multimedia files, source code, and so on). You use this dialog to attach the custom element form to the custom element.

Element	Description
Form Url	Specifies the custom element form that is used with the custom element.
OK	Saves your settings and closes the Custom Element Settings dialog.
Cancel	Cancels your settings and closes the Custom Element Settings dialog.
Help	Opens the online help for this specific dialog.

A.35 Static List Element Dialog

A static list combines the power and flexibility of a fragment with the contribution capabilities of an element in Contributor. A static list can bundle multiple elements in a table-like layout. Contributors can then add, edit, delete, and rearrange rows and columns of information. You can control the static list options that are available to a contributor using this dialog. This particular element is a legacy technique.

Element	Description
ID	The identification that Oracle Site Studio assigns to the element.
Name	<p>The name you assign to the element. This is used for coding purposes and as such should not contain spaces, or non-ASCII or special characters.</p> <p>If you reuse the same data file on multiple page templates, then the name of the region and its elements on each page must be the same.</p>
Label	Briefly describes the element. The label opens beside the field in Contributor to let individual contributors know what content should go there.
Info	Provides a more detailed description of the element. This description appears as a tooltip when contributors hover their mouse over the Info icon in Contributor.
Type	Indicates the type of element you are adding.

Element	Description
Actions	<p>Actions determine the available editing options for a contributor:</p> <p>Undo: Contributors can undo their last edit.</p> <p>Redo: Contributors can redo their last edit.</p> <p>Add Item: Contributors can add a row or column to the list.</p> <p>Remove Item: Contributors can remove a row or column from the list.</p> <p>Edit Existing Item: Contributors can edit a row or column in the list.</p> <p>Move Item Up: Contributors can move a row up in the list.</p> <p>Move Item Down: Contributors can move a row down in the list.</p> <p>Insert link: Contributors can create hyperlinks.</p> <ul style="list-style-type: none"> ▪ Link Settings: Opens the Link Settings dialog (see Section A.44, "Link Settings Dialog"), where you specify a contributor's linking options. <p>Remove link: Contributors can remove hyperlinks.</p> <p>Edit hyperlink targets: Contributors can edit the file that is the target of a hyperlink (using the Element menu and right-click menu).</p>
Actions (continued)	<p>Insert Image: Contributors can add/replace images (gif, jpg, and png).</p> <ul style="list-style-type: none"> ▪ Set QueryText: Opens the Edit Query Text dialog (see Section A.22, "Edit Query Text Dialog"), where you specify the query that is used in the content server to display images to the contributor. ▪ Show Results Only (On/Off): Shows or hides the content server environment when a contributor searches for images. <p>Apply CSS Classes: Contributors can apply different classes from a Cascading Style Sheet to text and graphics. If you enable this option, you should probably disable other formatting options (like font face, font size, bold, and italic) to more strictly enforce your CSS classes.</p> <ul style="list-style-type: none"> ▪ Set Class Names: Opens the Select CSS Classes dialog (see Section A.45, "Select CSS Classes Dialog"), where you specify the classes that is available in Contributor. <p>Validation: Used to validate the content added in Contributor (allowing you to enforce certain content requirements for contributors).</p> <ul style="list-style-type: none"> ▪ Validation Settings: Opens the Static List Element Validation dialog (see Section A.38, "Static List Element Validation Dialog"), where you specify what is validated. <p>Apply HTML Tags: Used to determine which tags are available to Contributors who edit in HTML.</p> <ul style="list-style-type: none"> ▪ Set Tag Names: Open to select specific tags that are allowed for Contributors. For more information, see Section A.50, "Select HTML Tags Dialog." <p>Set Element Height: Used to set a fixed height of the element.</p> <ul style="list-style-type: none"> ▪ Set Height: Use to set the height of the element by entering a number and the unit of measurement (centimeters, millimeters, inches, percent, picas, pixels, points). <p>Do Not Enclose Text in Editor: Used to define how carriage returns are handled in HTML. If text is enclosed, each time the contributor enters a return, a paragraph is created. If not enclosed, a line break is inserted for each return.</p>
Check All	Checks all of the actions so that a contributor has all available editing options.
Clear All	Clears all of the metadata values so that a contributor has no editing options.
OK	Saves your settings and closes the Static List Element dialog.
Cancel	Cancels your settings and closes the Static List Element dialog.

Element	Description
Help	Opens the online help for this specific dialog.

A.36 Static List Element Definition Dialog

A static list combines the power and flexibility that a fragment would have with the contribution capabilities of an element in Contributor. A static list can bundle multiple elements in a table-like layout. Contributors can then add, edit, delete, and rearrange rows and columns of information. You can control the static list options that are available to a contributor using this dialog.

Element	Description
Info	Provides a more detailed description of the element. This description appears as a tooltip when contributors hover their mouse over the Info icon in Contributor.
Type	Indicates the type of element you are adding.
Elements	Opens the Element Selection Screen (see Section A.37, "Elements Dialog")
Actions	<p>Actions determine the available editing options for a contributor:</p> <ul style="list-style-type: none"> Add Item: Contributors can add a row or column to the list. Remove Item: Contributors can remove a row or column from the list. Edit Existing Item: Contributors can edit a row or column in the list. Move Item Up: Contributors can move a row up in the list. Move Item Down: Contributors can move a row down in the list. <p>Validation: Used to validate the content added in Contributor (allowing you to enforce certain content requirements for contributors).</p> <ul style="list-style-type: none"> ▪ Validation Settings: Opens the Static List Element Validation dialog (see Section A.38, "Static List Element Validation Dialog"), where you specify what is validated. ▪ Set Element Height: Used to set a fixed height of the element. ▪ Set Height: Use to set the height of the element by entering a number and the unit of measurement (centimeters, millimeters, inches, percent, picas, pixels, points).
Check All	Checks all of the actions so that a contributor has all available editing options.
Clear All	Clears all of the metadata values so that a contributor has no editing options.
OK	Saves your settings and closes the Static List Element dialog.
Cancel	Cancel your settings and closes the Static List Element dialog.
Help	Opens the online help for this specific dialog.

The actions available in the static list element are for the list as a whole. The specific element definition options for the element definitions referenced within the static list are controlled within each element definition.

A.37 Elements Dialog

The Elements dialog is used to manage the elements associated with a static list. Other elements can be added, re-arranged, modified, or removed from the list.

Element	Description
List	List of named elements and their definitions associated with the region definition.
Add	Opens the Add Element dialog (see Section A.43, "Element Dialog"). From there you select an element to add to the list.
Remove	Removes the selected element from the list.
Edit	Opens the Add Element dialog (see Section A.43, "Element Dialog") to edit the selected element.
Move Up	Moves the selected element up the list.
Move Down	Moves the selected element down the list.
OK	Saves your settings and closes the Element Selection dialog.
Cancel	Cancels your settings and closes the Element Selection dialog.
Help	Opens the online help for this specific dialog.

A.38 Static List Element Validation Dialog

When contributors work in the Contributor application, they are generally free to add whatever content they like. As a designer, however, your goal may be to enforce certain standards (for example, limiting the number of rows a contributor can add to a static list). You can enforce this using element validation.

Element	Description
Use default validation	<p>Number Rows</p> <ul style="list-style-type: none"> ■ min: Enforces a minimum number of rows. ■ max: Enforces a maximum number of rows.
Use external validation	<p>Uses an external validation script that you provide.</p> <p>Settings: Opens the Advanced Element Validation dialog (see Section A.52, "Advanced Element Validation Dialog"), where you can enter the values from your script.</p>
OK	Saves your settings and closes the Static List Element Validation dialog.
Cancel	Cancels your settings and closes the Static List Element Validation dialog.
Help	Opens the online help for this specific dialog.

The default validation options in this dialog are derived from a script, *ss_default_validation_script.js*, that is provided by Oracle Site Studio.

A.39 Dynamic List Element Dialog

A dynamic list combines the power and flexibility of a fragment with contribution capabilities. The dynamic list performs a query in the content server, opens the files that match that query, and allows contributors to modify those files, even add new ones. You can control the contribution options that are available in the dynamic list using this dialog. This particular element is used in legacy sites, or in new sites as an old technique.

Element	Description
ID	The identification that Oracle Site Studio assigns to the element.
Name	The name you assign to the element. This is used for coding purposes and as such should not contain spaces, or non-ASCII or special characters. If you reuse the same data file in multiple places, then the name of the region and its elements on each page must be the same.
Label	Briefly describes the element. The label opens beside the field in Contributor to let individual contributors know what content should go there.
Info	A more detailed description that appears as a tooltip when contributors hover their mouse over the Info icon in Contributor.
Type	Indicates the type of element you are adding.
Actions	<p>Actions determine the available editing options for a contributor:</p> <p>Go to Next Page: Contributors can view the next set of matches in Contributor, if applicable.</p> <p>Go to Previous Page: Contributors can view the previous set of matches in Contributor, if applicable.</p> <p>Refresh: Contributors can refresh the dynamic list in Contributor. They likely do this after they add an item to the list.</p> <p>Create New Item: Contributors can add items (contributor data files or native documents) to the dynamic list.</p> <ul style="list-style-type: none"> ▪ New Item Settings: Opens the Region Content Options dialog (see Section A.20, "Region Content Options Dialog"), where you specify the type of files a contributor can add to the list. <p>Exclude Item: Contributors can remove items from the dynamic list. (The item is removed from all dynamic lists on the web site, but it is not removed from the content server.)</p> <p>Include Item: Contributors can include items in the dynamic list (that were previously excluded).</p>
Actions (continued)	<p>View Content Information for Item: Contributors can view the Content Information page for an item in the list.</p> <p>Edit Target Content Item: Contributors can edit the file after they add it to the list.</p> <p>Show Excluded Items: Contributors can view items not included in the Dynamic List and re-apply them to the list.</p> <p>Set Element Height: Used to set a fixed height of the element.</p> <ul style="list-style-type: none"> ▪ Set Height: Use to set the height of the element by entering a number and the unit of measurement (centimeters, millimeters, inches, percent, picas, pixels, points).
Check All	Checks all of the actions so that a contributor has all available editing options.
Clear All	Clears all of the metadata values so that a contributor has no editing options.
Parameters	Opens the Fragment Parameter Values dialog (see Section A.65, "Fragment Parameter Values Dialog"), where you can change the appearance and behavior of the dynamic list.
OK	Saves your settings and closes the Dynamic List Element dialog.
Cancel	Cancels your settings and closes the Dynamic List Element dialog.
Help	Opens the online help for this specific dialog.

When you allow contributors to add items to a list, they ultimately are adding web pages to the web site.

A.40 Dynamic List Element Definition Dialog

A dynamic list combines the power and flexibility that a fragment would have with contribution capabilities. The dynamic list performs a query in the content server, opens the files that match that query, and allows contributors to modify those files, even add new ones. You can control the contribution options that are available in the dynamic list using this dialog. This particular element is used in legacy sites.

Element	Description
Info	A more detailed description that appears as a tooltip when contributors hover their mouse over the Info icon in Contributor.
Type	Indicates the type of element you are adding.
Settings	Opens the Dynamic List Settings dialog. (see Section A.41, "Dynamic List Settings Dialog")
Actions	<p>Actions determine the available editing options for a contributor:</p> <p>Go to Next Page: Contributors can view the next set of matches in Contributor, if applicable.</p> <p>Go to Previous Page: Contributors can view the previous set of matches in Contributor, if applicable.</p> <p>Refresh: Contributors can refresh the dynamic list in Contributor. They likely do this after they add an item to the list.</p> <p>Create New Item: Contributors can add items (contributor data files or native documents) to the dynamic list.</p> <ul style="list-style-type: none"> ▪ New Item Settings: Opens the New Item Settings dialog (see Section A.53, "New Item Settings Dialog"), where you specify the type of files a contributor can add to the list. <p>Exclude Item: Contributors can remove items from the dynamic list. (The item is removed from all dynamic lists on the web site, but it is not removed from the content server.)</p> <p>Include Item: Contributors can include items in the dynamic list (that were previously excluded).</p> <p>View Content Information for Item: Contributors can view the Content Information page for an item in the list.</p> <p>Edit Target Content Item: Contributors can edit the file after they add it to the list.</p> <p>Show Excluded Items: Contributors can view items not included in the Dynamic List and re-apply them to the list.</p>
Actions	<p>Set Element Height: Used to set a fixed height of the element.</p> <ul style="list-style-type: none"> ▪ Set Height: Use to set the height of the element by entering a number and the unit of measurement (centimeters, millimeters, inches, percent, picas, pixels, points).
Check All	Checks all of the actions so that a contributor has all available editing options.
Clear All	Clears all of the metadata values so that a contributor has no editing options.
OK	Saves your settings and closes the Dynamic List Element dialog.
Cancel	Cancels your settings and closes the Dynamic List Element dialog.
Help	Opens the online help for this specific dialog.

When you allow contributors to add items to a list, they ultimately are adding web pages to the web site.

A.41 Dynamic List Settings Dialog

The Dynamic List Settings dialog is used to modify the settings for the dynamic list element definition. These settings would include the sort order and the specific query on the content server.

Element	Description
Query Text	The query on the content server that defines which items display in the dynamic list.
Edit	Opens the Edit Query Text dialog (see Section A.22, "Edit Query Text Dialog") to construct a query.
Sort Field	Select the field that serves as the sorting field for the returned search results.
Sort Order	Select to sort the search results in either ascending or descending order.
Number of results per page	Enter the number of search results returned per page.
Limit search scope to this website	Check to return only site assets which are associated with the web site.
Section to display result items	The target section in the web site where the results display.
Edit	Opens the Select Section dialog (see Section A.17, "Select Section Dialog") to select the target section.
OK	Saves your settings and closes the Dynamic List Settings dialog.
Cancel	Cancels your settings and closes the Dynamic List Settings dialog.
Help	Opens the online help for this specific dialog.

A.42 Region Definition Dialog

The region definition is used to define which elements, by the element definition, are available for use in a selected region template. The region definition is used to organize the set of element definitions available.

Element	Description
Description	The name of the region definition.
Elements	Makes select class names available in Contributor.
Add	Opens the Add Element dialog (see Section A.43, "Element Dialog") to add an element definition to the region definition.
Remove	Specifies a class name to be made available in Contributor. If you add a class name this way, you still must attach the CSS file containing this class to a page template or fragment.
Edit	Opens the Add Element dialog (see Section A.43, "Element Dialog") with the information for the selected element definition available to edit.
Move Up	Moves the selected element definition up the list.
Move Down	Moves the selected element definition down the list.
Modify Metadata	Opens the Enable Modify Metadata dialog (see Section A.15, "Enable Metadata Modification Dialog") to update Metadata fields for the region definition.

Element	Description
Switch Content	Opens the Region Content Options dialog (see Section A.20, "Region Content Options Dialog") to select the parameters contributors have to switch region content.
Help	Opens the online help for this specific dialog.

A.43 Element Dialog

The Element dialog is used to add an element definition to a region definition or a static list element definition. In this dialog you can add name-specific information to allow the element definition to maintain its reusability while still giving it a specific identifying name within the region definition or static list.

Element	Description
Name	The name of the element as it appears in the Designer view.
Label	The label of the element. The label serves as the name of the editing region in the contributor's editing region.
Info	Additional information about the element definition. The information entered here appears to the Contributor as a tooltip when the contributor hovers the mouse over the label.
Element Definition ID	Use to select the name of the element definition to use.
Embed the element definition inside the region definition	Check the box to embed the element definition in the region definition. This copies the element definition into the region definition, which means that the region definition must be updated when you want to update the element definition.
OK	Saves your settings and closes the Element dialog.
Cancel	Cancels your settings and closes the Element dialog.
Help	Opens the online help for this specific dialog.
Search	Opens a window displaying the results of a content server search for element definitions.
Edit	Opens a dialog to select the type of element definition to embed.

The info is what is listed in the hover on the tooltip in Contributor.

A.44 Link Settings Dialog

When you allow contributors to create links, they are able to use a link wizard (the same one that is available in Designer) to create a link to another section, to another site, or to another file. Using this dialog, you can control what types of files (contributor data files and native documents) the contributors can link to and whether they can link to new or existing files.

Element	Description
Create new contributor data file	Allows contributors to create a contributor data file that they can link to.

Element	Description
Create new native document	<p>Allows contributors to create a native document that they can link to.</p> <p>Document Types: Specifies which native documents contributors can create.</p> <p>The Additional Information button opens the Choose Document Types dialog (see Section A.21, "Choose Document Types Dialog"), where you can select the document types that are available to contributors.</p>
Choose existing in the content server	<p>Allows contributors to search the content server for an existing contributor data file or native document that they can link to.</p> <p>Query Text: Specifies the actual query that is performed in the content server to display existing files.</p> <p>The Additional Information button opens the Edit Query Text dialog (see Section A.22, "Edit Query Text Dialog"), which you can use to create, capture, and test the query used to display existing files.</p> <p>Show Results Only: Shows only the search results and not the typical content server environment. (This is useful to prevent contributors from browsing to other content in the content server.)</p>
Browse for content locally	Allows contributors to browse for a file on their local file system that they can link to.
Remove region content association	This option is disabled when you're specifying link settings for a contributor. It is only applicable when setting contribution region options for a contributor.
Default Metadata	Opens the Enable Metadata Modification dialog (see Section A.15, "Enable Metadata Modification Dialog"), where you can specify the metadata that is available and the default metadata that is used when contributors create files.
OK	Saves your settings and closes the Link Settings dialog.
Cancel	Cancels your settings and closes the Link Settings dialog.
Help	Opens the online help for this specific dialog.

A.45 Select CSS Classes Dialog

When you attach a cascading style sheet (CSS) to a page template, subtemplate, or region template, you can identify certain classes from the style sheet that are available to a contributor to choose from when editing text and graphics in Contributor. This can be useful when you want to enforce well-designed and consistent web pages. You can specify the class names that should be available in this dialog.

Element	Description
Allow all stylesheets	Makes all stylesheets within the scope of the current page available to Contributor, including any stylesheets referenced by fragments.
Allow selected stylesheets	Makes only the selected stylesheets available to Contributor.
Add ID...	Opens the Select a CSS Stylesheet dialog (see Section A.46, "Select a CSS StyleSheet Dialog") to select stylesheets from the content server.
Add Url...	Opens the Choose Fragment Asset dialog (see Section A.47, "Choose Fragment Asset Dialog") to select stylesheets referenced within fragments.
Remove	Removes the selected stylesheet from the stylesheets available to Contributor.
Allow all classes	Makes all classes from the selected stylesheets available in Contributor.

Element	Description
Allow selected classes	Makes select class names from the selected stylesheets available in Contributor.
Additional classes	Specifies a class name to be made available in Contributor. If you add a class name this way, you still must attach the CSS file containing this class to a page template or fragment.
	Creates a class name.
	Deletes the class name that you added (using the previous option).
OK	Saves your settings and closes the Select CSS Classes dialog.
Cancel	Cancels your settings and closes the Select CSS Classes dialog.
Help	Opens the online help for this specific dialog.

A.46 Select a CSS StyleSheet Dialog

This dialog is used to select a stylesheet from the content server. Once the stylesheet is selected the styles within the stylesheet will be accessible to Contributor.

Element	Description
Menu	Displays the content ID of the selected stylesheet.
Search...	Opens a search results window displaying all CSS files on the content server. From there, select the CSS file.
OK	Saves your settings and returns to the Select CSS Classes dialog.
Cancel	Cancels your settings and returns to the Select CSS Classes dialog.

A.47 Choose Fragment Asset Dialog

This dialog is used to select a stylesheet that is loaded with a fragment. Many of these stylesheets are not normally used except with fragments. Because of this, these fragments are loaded by Url rather than Content ID. Once the stylesheet is selected, the styles within the stylesheet will be accessible to Contributor.

Element	Description
Menu	Displays the available stylesheets loaded as fragment assets.
OK	Saves your settings and returns to the Select CSS Classes dialog.
Cancel	Cancels your settings and returns to the Select CSS Classes dialog.

A.48 WYSIWYG Element Validation Dialog

When contributors work in the Contributor application, they are generally free to add whatever text, graphics, and other content they like. As a designer, however, your goal may be to enforce consistency, especially formatting consistency, from one web page to another. You may also want to keep the content to a minimum on a particular page. You can enforce such standards with element validation (available in the WYSIWYG Element dialog).

Element	Description
Use default validation	<p>Uses the default validation options provided by Oracle Site Studio. Choose from the following:</p> <p>min: Enforces a minimum amount of text (in characters).</p> <p>max: Enforces a maximum amount of text (in characters).</p> <p>Disallowed characters: Prevents contributors from using the characters that you specify here.</p> <p>Allow line breaks
: When checked, contributors can create a line break (soft return), which is usually done by pressing SHIFT + ENTER on the keyboard.</p> <p>Allow paragraphs <p>: When checked, contributors can create a paragraph (hard return), which is usually done by pressing ENTER on the keyboard.</p> <p>Allow images : When checked, contributors can add images.</p>
Use external validation	<p>Uses an external validation script that you provide.</p> <p>Settings: Opens the Advanced Element validation dialog (see Section A.52, "Advanced Element Validation Dialog"), where you can enter the values from your script.</p>
OK	Saves your settings and closes the WYSIWYG Element Validation dialog.
Cancel	Cancels your settings and closes the WYSIWYG Element Validation dialog.
Help	Opens the online help for this specific dialog.

The default validation options in this dialog are derived from a script, *ss_default_validation_script.js*, that is provided by Oracle Site Studio.

A.49 Plain Text Element Validation Dialog

When contributors work in the Contributor application, they are generally free to add whatever text they like (even with the formatting limitations of a plain text element). As a designer, however, your goal may be to enforce consistency, especially if you must keep content to a minimum on a particular page. You can enforce such standards using element validation (available in the Plain Text Element dialog).

Element	Description
Use default validation	<p>Uses the default validation options provided by Oracle Site Studio. Choose from the following:</p> <p>min: Enforces a minimum amount of text (in characters).</p> <p>max: Enforces a maximum amount of text (in characters).</p> <p>Disallowed characters: Prevents contributors from using the characters that you specify here.</p> <p>Allow multiple lines: When checked, contributors can add lines of text (by pressing ENTER on their keyboard).</p>
Use external validation	<p>Uses an external validation script that you provide.</p> <p>Settings: Opens the Advanced Element Validation dialog (see Section A.52, "Advanced Element Validation Dialog"), where you can enter the values from your script.</p>
OK	Saves your settings and closes the Plain Text Element Validation dialog.
Cancel	Cancels your settings and closes the Plain Text Element Validation dialog.
Help	Opens the online help for this specific dialog.

The default validation options in this dialog are derived from a script, *ss_default_validation_script.js*, that is provided by Oracle Site Studio.

A.50 Select HTML Tags Dialog

When contributors are capable of editing a web page in the HTML rather than using a WYSIWYG interface, you may want to allow those contributors to edit specified regions using the HTML. As the designer, you can limit the available tags that the contributor can use in editing a region. This is helpful in maintaining a balance between the advanced abilities of some contributors and enforcing standards across a web page.

Element	Description
Allow default HTML tags	Allows the contributor to use the default set of HTML tags.
Allow selected HTML tags	Allows the contributor to use only those tags which are selected.
	Creates an allowed HTML tag.
	Deletes the HTML tag that you added (using the previous option).
OK	Saves your settings and closes the Plain Text Element Validation dialog.
Cancel	Cancels your settings and closes the Plain Text Element Validation dialog.
Help	Opens the online help for this specific dialog.

A.51 Image Element Validation Dialog

When contributors work in Contributor (in particular, with an image element), they are generally free to add whatever images they like. As a designer, however, your goal may be to enforce consistency, especially images sizes, from one page to another. You can enforce such standards using element validation (available in the Image Element dialog).

Element	Description
Use default validation	Uses the default validation options provided by Oracle Site Studio. Choose from the following: Width: <ul style="list-style-type: none"> ■ min: Enforces a minimum width (in pixels). ■ max: Enforces a maximum width (in pixels). Height: <ul style="list-style-type: none"> ■ min: Enforces a minimum height (in pixels). ■ max: Enforces a maximum height (in pixels).
Use external validation	Uses an external validation script that you provide. Settings: Opens the Advanced Element Validation dialog (see Section A.52, "Advanced Element Validation Dialog"), where you can enter the values from your script.
OK	Saves your settings and closes the Image Element Validation dialog.
Cancel	Cancels your settings and closes the Image Element Validation dialog.
Help	Opens the online help for this specific dialog.

If you enforce a certain image size in pixels, you may want to instruct contributors on how to change the size of the image using the image properties form.

The default validation options in this dialog are derived from a script, *ss_default_validation_script.js*, that is provided by Oracle Site Studio.

A.52 Advanced Element Validation Dialog

When contributors work in Contributor, they are generally free to add whatever text, graphics, and other content they like. As a designer, however, your goal may be to enforce consistency or to limit the amount of content that is contributed to a web page. You can use the default validation options in Designer or come up with your own validation script and attach it to the element using this dialog.

Element	Description
Script	Specifies the custom validation script that you use for the element. The Additional Information button opens an Oracle Content Server Dialog where you can select your validation script from the content server.
Language	Specifies the scripting language used in your validation (JavaScript or VBScript).
Method Name	Specifies the method that you use to run the validation.
Content Type	Specifies the content type (string , XML , HTML) that is used in the validation.
	Adds a parameter.
	Deletes an existing parameter.
	Moves the parameter up in the list.
	Moves the parameter down in the list.
OK	Saves your settings and closes the Advanced Element Validation dialog.
Cancel	Cancels your settings and closes the Advanced Element Validation dialog.
Help	Opens the online help for this specific dialog.

A.53 New Item Settings Dialog

When you allow contributors to add an item to a dynamic list, they have the ability to add a new or existing contributor data file or native document. You can control exactly what types of files they can add using this dialog (accessible from the Dynamic List Element Definition dialog).

Element	Description
Create new contributor data file	Allows contributors to create a contributor data file and add it to the list.
Create new native document	Allows contributors to create a native document and add it to the list. Document Types: Specifies which native documents contributors can create. The Additional Information button opens the Choose Document Types dialog (see Section A.21, "Choose Document Types Dialog"), where you can select the document types that are available to contributors.

Element	Description
Choose existing in the content server	This option is disabled in the context of a dynamic list because the query for the dynamic list already picks up existing files in the content server. If a contributor wants to add an existing file in the content server that doesn't already appear in the list, then the metadata for that file must be changed to match the metadata used in the list.
Browse for content locally	Allows contributors to browse for a file on their local file system and add it to the list.
Remove region content association	This option is disabled in the context of a dynamic list. It is only applicable when setting contribution region options for a contributor.
Default Metadata	Opens the Enable Metadata Modification dialog (see Section A.15, "Enable Metadata Modification Dialog"), where you can specify the metadata that is available and the default metadata that is assigned to the files added to the list.
OK	Saves your settings and closes the New Item Settings dialog.
Cancel	Cancels your settings and closes the New Item Settings dialog.
Help	Opens the online help for this specific dialog.

A.54 Assign Content Dialog

Each placeholder on a primary page needs a contributor data file, native document, or subtemplate assigned to it. Each placeholder on a secondary page needs a subtemplate, data file, or native document assigned to it, or it must be designated as replaceable. You perform these tasks in this dialog. The contributor can also assign data files and edit the data files assigned to a placeholder (assuming you allow these actions).

Element	Description
Site Hierarchy	Displays the site hierarchy and indicates which placeholders have content assigned to them. The folder icon indicates the section has content assigned or is designated as replaceable. The X-folder icon indicates the section is missing content or must be designated as replaceable. Click a section to view and configure the placeholder (or placeholders) on the primary and secondary pages there.
Primary	Displays the primary pages in the site hierarchy.
Secondary	Displays the secondary pages in the site hierarchy.
Show all	Expands the site hierarchy and displays the sections that are missing content or are not designated as replaceable. If you have more than one, you see a navigation control (see below) that you can use to go through each one.
Finished	Saves your settings and closes the Assign Content dialog.
Help	Opens the online help for this specific dialog.

You can choose not to assign content to a region and instead have a contributor perform this task.

Placeholder Information Tooltip

When the contribution graphic opens in the Assign Region dialog, you can hover over the contribution graphic for the placeholder and when the region highlighting appears, a placeholder information tooltip also opens in a small popup window.

The placeholder information tooltip lists the name of the placeholder mapping, placeholder definition, region definition, and region template that have been assigned in the highlighted contribution region. The Content ID of the page template is also displayed. This information can help in remembering what site assets have been assembled in each placeholder.

A.55 Choose Region Content Dialog

Each region on a primary page needs a contributor data file or native document assigned to it. Each region on a secondary page needs a file assigned to it, or it must be designated as replaceable. You can assign a new or existing contributor data file or native document to a region using this dialog, accessible from the Assign Content Dialog (see [Section A.54, "Assign Content Dialog"](#)).

Element	Description
Content ID	Displays the Content ID of the file assigned to the region. (This field and the ones below are initially blank until you assign a file to the region.)
Title	Displays the Title of the file assigned to the region.
Type	Displays the Type (document type) of the file assigned to the region.
Author	Displays the Author of the file assigned to the region.
Comments	Displays the Comments of the file assigned to the region.
Document Info	Opens the Content Information page for the file assigned to the region.
Choose	Use to select a file that you assign to the region. You have four choices: Existing: Opens the Oracle Content Server dialog where you can select an existing contributor data file or native document and assign it to the region. New: Displays a list of file types that you can create and assign to the region. Local: Opens a window that you use to select a file from your local file system, checked it in, and assign it to the region. None: Removes the existing file assigned to the region.
OK	Saves your settings and closes the Choose Region Content dialog.
Cancel	Cancels your settings and closes the Choose Region Content dialog.
Help	Opens the online help for this specific dialog.

A.56 Generate Unique Region Content Dialog

Oracle Site Studio can automatically generate a unique contributor data file for each region on the same page template throughout the site hierarchy. This is useful when you reuse a page template and want to populate the contribution region on that page wherever it is used on your site. You can specify your own naming convention when you use this feature.

Contributor data files already assigned to a region are not overwritten.

Element	Description
Use auto-generated Content IDs	The Content ID is automatically generated based on your settings in the content server. This option is disabled if the content server is not set up to auto-generate content IDs.
Create Content IDs from the following	The Content ID is automatically generated based on the naming convention that you specify here: Prepend with: This prefix is used followed by the naming convention specified below: <ul style="list-style-type: none"> ■ Use timestamp to ensure uniqueness: The name is based on a unique timestamp (when the file was created and checked into the content server). ■ Use section and region to ensure uniqueness: The name is based on the section and region where the file is located on the web site.
Define	Defines additional default metadata values for each file by using the Assign Info Form in the content server.
Auto-generate title	Automatically generates a title based on the name of the section and the name of the region. The title, for example, may look similar to this: Contribution Data For Section 'Products', Region 'Region1'
Auto-generate comments	Automatically generates a comment based on the name of the section and the name of the region. The comments, for example, may look something like this: Comments: This is region content for the Oracle Site Studio Web Site Section 'Products', Region 'Region1'
OK	Saves your settings and closes the Generate Unique Region Content dialog.
Cancel	Cancels your settings and closes the Generate Unique Region Content dialog.
Help	Opens the online help for this specific dialog.

A.57 Fragment Editor Dialog

The Fragment Editor comprises a suite of editing options that allow you to edit all parts of a fragment, including its snippets, assets, parameters, and elements. You can open the Fragment Editor from the File menu, and whenever you edit a fragment in the toolbox.

Element	Description
	Changes the icon associated with the fragment. The icon displays in the Toolbox after you save the fragment. Click the button to see the available icons and then select one of them.
	Opens the Properties for Fragment dialog (see Section A.58, "Properties for Fragment Dialog"), where you can change the fragment's name, language, and type.
Assets	Opens the Fragment Assets dialog (see Section A.59, "Fragment Assets Dialog"), where you can add, edit, and delete the assets in a fragment.

Element	Description
Elements	<p>Opens the Fragment Elements dialog (see Section A.60, "Fragment Elements Dialog"), where you can add, edit, and delete each element in the fragment.</p> <p>Elements are only available for static list fragments. As such, this button is disabled if the fragment type is not "staticlist" (see Section A.58, "Properties for Fragment Dialog").</p>
Parameters	<p>Opens the Fragment Parameters dialog (see Section A.61, "Fragment Parameters Dialog"), where you can add, edit, and delete each parameter in the fragment.</p>
Add	<p>Opens the Snippet Properties dialog (see Section A.64, "Snippet Properties Dialog"), where you can add a snippet to the fragment.</p>
Delete	<p>Deletes the snippet from the fragment.</p>
Properties	<p>Opens the Snippet Properties dialog (see Section A.64, "Snippet Properties Dialog"), where you can edit a snippet in the fragment.</p>
Name Location Include (Status area)	<p>Displays the existing snippets in the fragment.</p>
Snippet code	<p>Displays the snippet code. You can edit the text in this window.</p>
Preview	<p>Opens the Fragment Preview dialog (see Section A.66, "Fragment Preview Dialog"), where you can preview the fragment.</p> <p>You may be prompted to choose a set of parameter values to preview the fragment.</p>
Help	<p>Opens the online help for this specific dialog.</p>

A.58 Properties for Fragment Dialog

The properties of a fragment include the fragment's name, ID, language, and type. Each time you create a fragment, you're prompted to enter these values. You can also change these values after you create a fragment using this dialog.

Element	Description
Name	<p>Identifies the fragment.</p> <p>The name of the fragment displays in the Toolbox. It can be a friendly name that includes spaces and special characters, if you like.</p>
ID	<p>Identifies the fragment and its individual parts (snippets, assets, parameters, and so on).</p> <p>You can only change the ID when you first create the fragment. The ID also serves as an XML attribute and as the base name for the classes, files, and directories in the fragment. The ID should be concise, and it should not contain spaces, or non-ASCII or special characters.</p>
Language	<p>Identifies the language used by the fragment for server-side scripting.</p> <p>If your site type is "hcsp/jsp," then the available language is idoc or jsp. If your site type is "asp," then the available language is asp.</p>

Element	Description
Type	<p>Identifies the type and intended use of the fragment. This determines the available options for the fragment and its location in the Toolbox. There are four types:</p> <p>navigation: The fragment is used for site navigation (navigation bar, breadcrumbs, search, and so on).</p> <p>staticlist: The fragment is used to provide advanced contribution capabilities, where contributors can edit individual elements.</p> <p>dynamic list: The fragment is used to provide advanced contribution capabilities, where contributors can modify files that appear in the list.</p> <p>other: The fragment is used for miscellaneous purposes, such as a copyright line, embedded multimedia, or a login script.</p>
Library Name	<p>Identifies the library that the fragment is stored in.</p> <p>Initially, it says "<unsaved>" and appears disabled, until you save the fragment for the first time.</p>
OK	Saves your settings and closes the Fragment Properties dialog.
Cancel	Cancels your settings and closes the Fragment Properties dialog.
Help	Opens the online help for this specific dialog.

A.59 Fragment Assets Dialog

Assets are files that are referenced by a fragment, usually within a fragment snippet. A graphic, a cascading style sheet (CSS), and a standalone script are all frequently used assets. You can add, edit, and delete assets in this dialog.

Element	Description
Current Assets	Displays existing fragment assets.
Add Asset	<p>Adds an asset to the fragment.</p> <p>You add assets from your file system. Oracle Site Studio then checks the file into the content server and packages it with this fragment.</p>
Add Folder	<p>Adds a subfolder for the assets, which you can use to better organize the files.</p> <p>To ensure compatibility on all platforms (specifically UNIX), you should use all lowercase letters in the subfolder.</p>
Delete	Deletes the asset from the fragment.
Edit	<p>Edits the asset (using a third-party application). There are two options available.</p> <p>Edit: Opens the asset with the application associated with that file type.</p> <p>Edit With: Opens a dialog where you choose an application to open the file with.</p>
OK	Saves your settings and closes the Fragment Assets dialog.
Cancel	Cancels your settings and closes the Fragment Assets dialog.
Help	Opens the online help for this specific dialog.

A.60 Fragment Elements Dialog

An element becomes a field in Contributor where users can add and edit content (text and graphics). Elements can also be bundled in a fragment to create a static list. With

static lists, contributors can add, edit, delete, and rearrange rows and columns of information. You can add, edit, and delete elements in this dialog.

Element	Description
	Opens the WYSIWYG Element dialog (see Section A.26, "WYSIWYG Element Dialog"), where you add the element to the fragment.
	Opens the Plain Text Element dialog (see Section A.28, "Plain Text Element Dialog"), where you add the element to the fragment.
	Opens the Image Element dialog (see Section A.30, "Image Element Dialog"), where you add the element to the fragment.
	Opens the Custom Element dialog (see Section A.32, "Custom Element Dialog"), where you add the element to the fragment.
Edit	Edits the selected element in the fragment.
Delete	Deletes the selected element in the fragment.
Name Label Type Description	Displays existing elements in the fragment.
OK	Saves your settings and closes the Fragment Elements dialog.
Cancel	Cancel your settings and closes the Fragment Elements dialog.
Help	Opens the online help for this specific dialog.
Advanced	Opens a text editor that you use to customize your element even more. When you edit an element this way, you are not able to return to the graphical user interface in the Fragment Elements dialog because there is no supported interface available.

A.61 Fragment Parameters Dialog

Parameters are attributes that govern the appearance and behavior of a fragment when it is added to a page template. You create parameters in this dialog and then choose from those parameters when you use the fragment (see [Section A.65, "Fragment Parameter Values Dialog"](#)).

Element	Description
Parameters: Name	Displays the existing parameters in the fragment.
	Creates a parameter in the fragment. The name is used for coding purposes and as such, should not contain spaces, or non-ASCII or special characters. Also, you should not use words that are commonly used for coding purposes, such as "like" (used by Idoc Script as a string comparison operator).
	Deletes the parameter in the fragment.
	Moves a parameter up in the list. This changes its location in this dialog and the Fragment Parameter Values dialog (see Section A.65, "Fragment Parameter Values Dialog").
	Moves a parameter down in the list. This changes its location in this dialog and the Fragment Parameter Values dialog (see Section A.65, "Fragment Parameter Values Dialog").

Element	Description
Type	<p>Defines the function, or purpose, of the parameter. There are several types:</p> <p>text: Provides a simple text box where you can insert text when using the fragment. (You can create an option list for the text using the Option List: Label Value)</p> <p>bigtext: Provides a much larger text box (in a popup window) where you can add and edit text when using the fragment.</p> <p>boolean: Provides a True or False value that you can specify when using the fragment.</p> <p>integer: Provides an integer value that you can specify when using the fragment. (This value may not contain decimal points.)</p> <p>float: Provides a value, including decimal points, that you can specify when using the fragment.</p> <p>size: Provides a sizing value, in pixels or percentage, that you can specify when using the fragment.</p>
Type (continued)	<p>color: Provides an RGB hexadecimal value (for example, 0xFF0000) and a color picker that you can use to select a color with when using the fragment.</p> <p>url: Provides a dialog that you can use to browse to a URL and then select it as a value when using the fragment.</p> <p>manageddoc: Opens a content server window that you use to browse to a document in the content server and select it as a value (dDocName), when using the fragment. (Use the Query Text field below to capture the query)</p> <p>managedurl: Opens a content server window that you use to browse to the URL of a document in the content server and then select it as a value, when using the fragment. (Use the Query Text field below to capture the query)</p> <p>managedquery: Opens the Edit Query Text dialog (see Section A.22, "Edit Query Text Dialog"), where you can build a query, when using the fragment. This parameter is most useful for dynamic list fragments.</p> <p>cssstyle: Provides a text box where you can specify a CSS style value when using the fragment.</p> <p>siteid: Provides a dialog (see Section A.16, "Select Site Dialog") that you use to select a web site in the content server, when using the fragment.</p> <p>nodeid: Opens a dialog (see Section A.17, "Select Section Dialog") that you use to select a section in your site hierarchy, when using the fragment.</p> <p>custom: Provides advanced functionality for the fragment parameter. (See <i>Technical Reference for Oracle Site Studio.</i>)</p>
Required	<p>Makes the parameter required.</p> <p>When using the fragment, you must enter values for each parameter before the Fragment Parameter Values dialog (see Section A.65, "Fragment Parameter Values Dialog") closes.</p> <p>You can make a parameter required and also supply an acceptable default value.</p>
Description	<p>Provides a description of the parameter.</p> <p>The description displays at the bottom of the Fragment Parameter Values dialog when you use the fragment.</p>
Default value	<p>Provides a default value for the parameter when you actually use the fragment.</p>

Element	Description
Query Text	Specifies the query that is used when you browse for a document using the "manageddoc" or "managedurl" parameter type (above). If no query is specified, everything is searched in the content server. The Additional Information button opens the Edit Query Text dialog (see Section A.62, "Edit Query Text Dialog"), where you can enter your query, capture it, and test it in the content server.
Option List: Label Value	Displays existing option lists in the fragment. Option lists are used to specify a list of possible values for a parameter of type "text" (above). The values display as a popup list of options in the Fragment Parameter Values dialog when you use the fragment.
Add	Opens the New Option dialog (see Section A.63, "New Option Dialog"), where you can add an option list.
Edit	Opens the New Option dialog (see Section A.63, "New Option Dialog"), where you can edit an option list.
Delete	Deletes an option list.
OK	Saves your settings and closes the Fragment Parameters dialog.
Cancel	Cancels your settings and closes the Fragment Parameters dialog.
Help	Opens the online help for this specific dialog.

A.62 Edit Query Text Dialog

When you add a parameter to a fragment that is used to search the content server (for example, parameters of type manageddoc, managedurl, and managedquery), you must specify the query that is performed. You can use this dialog to specify the query, capture the query, and test the query.

Element	Description
[Text field]	Use this text box to specify your query.
Capture Query	Opens the Content Query Capture screen in the content server that you use to perform a query and then save (capture) that query.
Test Query	Use to test your query in the content server.
OK	Saves your settings and closes the Edit Query Text dialog.
Cancel	Cancels your settings and closes the Edit Query Text dialog.

A.63 New Option Dialog

When you create a parameter of type "text" in a fragment (see [Section A.61, "Fragment Parameters Dialog"](#)), you can create a list of values in the form of an option list. These values display as a popup list in the Fragment Parameter Values dialog when the fragment is added to a page template. You can create the option list in this dialog.

Element	Description
Label	Displays the name in the option list.
Value	Displays the value in the option list (what is added to the page template).
OK	Saves your settings and closes the New option dialog.
Cancel	Cancels your settings and closes the New option dialog.

Element	Description
Help	Opens the online help for this specific dialog.

A.64 Snippet Properties Dialog

A snippet is the text or code that a fragment adds to a page template (in one of four locations). A fragment must have at least one snippet. You can define the properties of the snippet in this dialog, and then edit the content of the snippet in the Fragment Editor.

Element	Description
Name	The name of the snippet. (The name may contain spaces and special characters, if you like.)
Location	Determines the location where the snippet is added on a page template. There are four options: drop-point: Inserts the snippet where the cursor is located on the page template. head: Inserts the snippet in the head of the page template (specifically the end of the <HEAD> tag). top-of-body: Inserts the snippet at the beginning of the body in the page template (immediately following the <BODY> tag). bottom-of-body: Inserts the snippet at the end of the body in the page template (immediately preceding the </BODY> tag).
Include	Determines the way in which the snippet is added to a page template. There are three options: simple: Adds the entire snippet to the page template. If parameters are present, their values are added directly to the page, making them a permanent part of the page. The fragment is no longer recognized as a fragment and its parameters cannot be changed. inline: Adds the entire snippet to the page template. As an inline snippet, however, the fragment and its parameters are still recognized as a fragment. You can still move the fragment, delete it, and modify its parameters. reference: Adds a reference to the snippet in the page template (similar to an include file). As a referenced snippet, it is still recognized and managed as a fragment, and you can still edit the fragment in the Fragment Editor and then see the changes on your page template.
Design View Code	Creates placeholder text for the fragment snippet when the snippet is viewed in Design view. The Additional Information button opens a text editor that you use to enter placeholder text. This Element is intended for snippets inserted in the drop-point, top-of-body, and bottom-of-body locations; not the head location.
OK	Saves your settings and closes the Snippet Properties dialog.
Cancel	Cancels your settings and closes the Snippet Properties dialog.
Help	Opens the online help for this specific dialog.

A.65 Fragment Parameter Values Dialog

Parameters are attributes that govern the appearance and behavior of a fragment when it is added to a page template. You can create parameters in the Fragment Editor (see

[Section A.61, "Fragment Parameters Dialog"](#)) and then choose from those parameters in this dialog, when you're using the fragment.

Parameters are optional, and therefore not all fragments have them.

Element	Description
Property	Lists the available parameters in the fragment. Each fragment contains different parameters (some contain no parameters at all) and as such, the parameters vary from fragment to fragment.
Name	Lists the values for each parameter.
	Creates a parameter in the fragment. The name is used for coding purposes and as such, should not contain spaces, or non-ASCII or special characters. Also, you should not use words that are commonly used for coding purposes, such as "like" (used by Idoc Script as a string comparison operator).
	Deletes the parameter in the fragment.
	Moves a parameter up in the list. This changes its location in this dialog and the Fragment Parameter Values dialog (see Section A.65, "Fragment Parameter Values Dialog").
	Moves a parameter down in the list. This changes its location in this dialog and the Fragment Parameter Values dialog (see Section A.65, "Fragment Parameter Values Dialog").
OK	Saves your settings and closes the Fragment Parameter Values dialog.
Cancel	Cancels your settings and closes the Fragment Parameter Values dialog. The fragment is not added to the page template.
Help	Opens the online help for this specific dialog.

You can modify the available parameters, add parameters, and change the default values using the "Fragment Parameters dialog."

A.66 Fragment Preview Dialog

You can preview fragments before adding them to a page template. You can open this dialog by right-clicking a fragment in the Toolbox and then choosing Preview or use the Preview button in the Fragment Editor. You may be prompted to first choose a set of parameter values before you can preview the fragment (see [Section A.65, "Fragment Parameter Values Dialog"](#)).

Element	Description
Select section to use for preview	Displays the site hierarchy. The section that you select is used to display the fragment in the Fragment Preview area. You are not able to select a section that is excluded from the site navigation, has no primary page, or contains an external URL as the primary page.
Fragment Preview	Previews the fragment using the section that is highlighted in the site hierarchy (above). You cannot preview static list fragments because they require a contributor data file with content in it to be viewed properly.

Element	Description
Change Parameter Values	Opens the Fragment Parameter Values dialog (see Section A.65, "Fragment Parameter Values Dialog"), where you can change the parameter values used in the preview.
Finished	Closes the Fragment Preview dialog.
Help	Opens the online help for this specific dialog.

A.67 Save Fragment Dialog

Fragments are stored in fragment libraries. You can save a fragment in the same library as other fragments, or you can save the fragment in its own library. You can also create a library for the fragment in this dialog.

Element	Description
New	Creates a library for the fragment. The library is created in the content server, and it opens in the Save Fragment dialog the next time you open it.
Name	Displays existing fragment libraries in the content server. Select one of them to store the fragment in that library.
OK	Closes the Save Fragment dialog and takes you to the Assign Info form in the content server where you can update the metadata for the fragment library.
Cancel	Closes the Save Fragment dialog without saving your fragment.
Help	Opens the online help for this specific dialog.

The Save Fragment dialog only appears the first time you save a fragment. For each subsequent save, you're taken directly to the Assign Info Form for the fragment library in the content server.

A.68 Manage Site Page

Using Manager, you can modify all aspects of your site hierarchy. You can add and remove sections, assign a different page template to a section, change the properties associated with a section, and so on.

Element	Description
(Manage Site) (Refresh)	"Manage Site" displays when Manager opens in a collapsed state. Click the link to expand it. "Refresh" displays when Manager is in an expanded state. Click the link to refresh Manager and see your changes.
Site Hierarchy	Displays the site hierarchy and is used to modify the site hierarchy. The tasks performed here require the mouse and at times a right-mouse click.
Section	Use to make common site hierarchy changes, like adding a section, removing a section, and changing the properties of a section.
Layout	Use to preview a primary page, assign a different primary page (or external URL), and view the content information page for a primary page.

Element	Description
Secondary Layout	Use to preview a secondary page, assign a different secondary page, and view the content information page for a secondary page.
Custom Properties	Use to view and change the custom properties for a section. Custom properties are created by the designer and typically apply to this one site or related sites on the content server.

A.69 Manager Configuration Settings Dialog

The manager configuration settings dialog is used to customize the manager application as viewed from within the Contributor interface.

Element	Description
Visible in Contributor Mode only	Check to make the Manager interface available only when the page is viewed in Contributor mode.
Automatically load Manager	Check to automatically load the Manager fully when the page loads.
Hierarchy	Select to display the complete hierarchy, only the section the manager component appears in, or none of the hierarchy when the Manager application loads.
Add New Section	Check to allow a section to be added.
Set UrlDirName	Check to allow the UrlDirName value to be added when a section is created.
Set Section ID	Check to allow the Section ID value to be added when a section is created.
Remove Section	Check to allow a section to be removed.
Move Section	Check to allow a section to be moved.
Set the Error Handler Section	Check to allow a section to be marked as the Error Handler.
Change the Primary Page Template	Check to allow a Manager to change the page template used for the primary age of the section.
Change the Secondary Page Template	Check to allow a Manager to change the page template used for the secondary age of the section.
Section ID	Check to display the Section ID field for a section. disabled: makes the value read-only.
Label	Check to display the Label field for a section. disabled: makes the value read-only.
Include in Navigation	Check to display the Include in Navigation field for a section. disabled: makes the value read-only.
Contributor Only	Check to display the Contributor Only field for a section. disabled: makes the value read-only.
UrlDirName	Check to display the UrlDirName field for a section. disabled: makes the value read-only.
UrlPageName	Check to display the UrlPageName field for a section. disabled: makes the value read-only.

Element	Description
MaxAge	Check to display the MaxAge field for a section. disabled: makes the value read-only.
Custom	Check to display the Custom Properties tag on the Manger interface.
OK	Closes the Save Fragment dialog and takes you to the Assign Info form in the content server where you can update the metadata for the fragment library.
Cancel	Closes the Save Fragment dialog without saving your fragment.
Help	Opens the online help for this specific dialog.

A.70 Placeholder Section Properties Dialog

The Placeholder Section Properties dialog lists the placeholder mappings used within the section that the dialog was opened from.

Element	Description
Name	The name of the placeholder.
Data File ID	Specifies the ID of the content (contributor data file or native document) mapped to the named placeholder section property on the page.
Region Template or Subtemplate	Specifies the region template or subtemplate assigned to the placeholder.
Placeholder Definition ID	Specifies the ID of the placeholder definition assigned to the placeholder.
Add	Opens the Edit Placeholder Section Properties dialog (see Section A.71, "Edit Placeholder Section Properties Dialog") to add a mapping to the list.
Remove	Removes the selected placeholder definition mapping from the list.
Edit	Opens the Placeholder Definition Mapping dialog (see Section A.72, "Placeholder Definition Dialog") with the values of the selected mapping to edit.
OK	Saves your settings and closes the Placeholder Section Properties dialog.
Cancel	Cancel your settings and closes the Placeholder Section Properties dialog.
Help	Opens the online help for this specific dialog.

A.71 Edit Placeholder Section Properties Dialog

With this dialog, you define the placeholder section properties.

Element	Description
Name	The name of the placeholder.
Datafile	The contributor data file or native document for this placeholder.

Element	Description
Region Template or Subtemplate	The region template or subtemplate for this placeholder.
ID	The placeholder definition for this placeholder.
Search	In each instance, opens a content server search results page that opens results appropriate to the field.
OK	Saves your settings and closes the Edit Placeholder Definition Mappings dialog.
Cancel	Cancels your settings and closes the Edit Placeholder Definition Mappings dialog.
Help	Opens the online help for this specific dialog.

A.72 Placeholder Definition Dialog

The placeholder definition is used to control how the site assets within the placeholder are used and how they work. The placeholder is also used to control the workflow, site reports, and whether the contributor can even edit the data within the placeholder.

Element	Description
Description	The name of the placeholder definition.
Contributor edit	Check to allow the contributor to have access to edit the contributor data file within the region.
Document info	Check to make the document info menu available on the document info page.
Update document info	Check to allow the contributor access to edit the document info. It is not necessary to select Document Info to allow the contributor access to edit.
Workflow approve	Check to include workflow approval in the contributor menu. This is only valid when the item is in workflow and the contributor can approve the item in workflow.
Workflow reject	Check to include workflow reject in the contributor menu. This is only valid when the item is in workflow and the contributor can reject the item in workflow.
Modify metadata	Check to allow the contributor access to edit the metadata.
Switch content	Check to allow the contributor access to switch the contributor data file or the native document.
Switch region template	Check to allow the contributor access to switch the region template.
View Web site usage report	Check to allow the contributor access to view the web site usage report.
Remove content	Check to allow the contributor access to remove the content (by removing the association with the contributor data file or subtemplate) displayed in the placeholder.
Allowed region definitions	The list of region definitions associated with the placeholder definition. The region definitions listed here are the only subtemplates available to use within the placeholder the placeholder definition is mapped to.

Element	Description
Associated region templates	The list of all region templates associated with the region definition selected in the "Allowed region definitions" list. The region templates listed here are the only region templates available to use within the placeholder the placeholder definition is mapped to. The region template marked with an asterisk is the default region template for the selected region definition.
Allowed subtemplates	The list of all region templates associated with the region definition selected in the "Allowed region definitions" list. The subtemplates listed here are the only subtemplates available to use within the placeholder the placeholder definition is mapped to.
Add	Click to add a region definition, region template, or subtemplate to the placeholder definition.
Remove	Click to remove a region definition, region template, or subtemplate to the placeholder definition.
Make Default	Click to mark a region template as the default template for the selected region definition
Help	Opens the online help for this specific dialog.

A.73 Placeholder Definition Mappings Dialog

This dialog is used to organize and define the placeholders and their definitions when used on primary and secondary pages. Since a placeholder can appear on a page template that is used both as a primary and a secondary page, it can be used to define different rules depending on where the placeholder is used.

Element	Description
Name	The name of the placeholder.
Definition ID (Primary)	Specifies the ID of the placeholder definition to use with the placeholder when the page template is used as a primary page.
Definition ID (Secondary)	Specifies the ID of the placeholder definition to use with the placeholder when the page template is used as a secondary page.
Add	Opens the Placeholder Definition Mapping dialog (see Section A.74, "Define Placeholder Definition Mapping Dialog") to add a mapping to the list.
Remove	Removes the selected placeholder from the list.
Edit	Opens the Placeholder Definition Mapping dialog (see Section A.74, "Define Placeholder Definition Mapping Dialog") with the values of the selected mapping to edit.
OK	Saves your settings and closes the Placeholder Definition Mappings dialog.
Cancel	Cancels your settings and closes the Placeholder Definition Mappings dialog.
Help	Opens the online help for this specific dialog.

A.74 Define Placeholder Definition Mapping Dialog

This dialog is used to define which placeholder definitions are used when the placeholder is on a primary page, or on a secondary page.

Element	Description
Name	Enter the name of the placeholder.
Definition ID (Primary)	Specifies the ID of the placeholder definition to use with the placeholder when the page template is used as a primary page.
Definition ID (Secondary)	Specifies the ID of the placeholder definition to use with the placeholder when the page template is used as a secondary page.
Search	Displays the results of a search on the content server for placeholder definitions.
OK	Saves your settings and closes the Define Placeholder Definition Mapping dialog.
Cancel	Cancels your settings and closes the Define Placeholder Definition Mapping dialog.
Help	Opens the online help for this specific dialog.

A.75 Define Placeholder Dialog

The Define Placeholder dialog is used to select a placeholder name.

Element	Description
Name	Use to select the placeholder by name.
Optionally specify the placeholder definition on the template	Marks the placeholder definition within the template, rather than using a mapping to reference the placeholder definition as a site asset.
OK	Saves your settings and closes the Define Placeholder dialog.
Cancel	Cancels your settings and closes the Define Placeholder dialog.
Help	Opens the online help for this specific dialog.

A.76 Switch Content Wizard

The Switch Content wizard is used to change the content assigned to a placeholder (or assign content if none was assigned yet). The content currently there can be replaced with a contributor data file, a native document, or even a subtemplate.

The Switch Content wizard contains these screens:

- [Section A.76.1, "Switch Content Wizard: Choose Region Content Screen"](#)
- [Section A.76.2, "Switch Content Wizard: Choose Content File Screen"](#)
- [Section A.76.3, "Switch Content Wizard: Check-In Content Screen"](#)
- [Section A.76.4, "Switch Content Wizard: Confirmation Screen"](#)

A.76.1 Switch Content Wizard: Choose Region Content Screen

In the first screen of the Switch Content wizard, you are asked to pick either a file based on an association with a region definition, or a subtemplate.

Element	Description
Select content based upon this Region Definition	Selects an available region definition that the content is associated with. If you want to create a new file or use a local file, the content checked in is associated with this region definition.
Select a Subtemplate	Selects a subtemplate that the content is associated with.
Cancel	Cancels your settings and closes the Switch Content wizard.
Back	Disabled on this screen.
Next	Takes you to the next screen in the wizard.
Finish	Disabled on this screen.
Help	Opens the online help for this specific dialog.

If you select a subtemplate, the wizard completes. If you select content, then you continue in the wizard.

A.76.2 Switch Content Wizard: Choose Content File Screen

After selecting a region definition, the content you switch to can be either located on the content server already, or you can create a new piece of content, or you can also check in an existing local file to the content server and use that.

Element	Description
New Contributor data file	Select to create and check in a new contributor data file.
New native file	Select to create and check in a new native document of the type chosen in the menu.
Existing file from server	Select to open a search results window.
Existing local file	Select to check in and upload a file to the content server that is located on the local computer.
None	Select to remove the current content.
Current item	Select to keep the current item. The current item's information is also displayed.
Cancel	Cancels your settings and closes the Switch Content wizard.
Back	Takes you to the previous screen.
Next	Takes you to the next screen in the wizard.
Finish	Disabled on this screen.
Help	Opens the online help for this specific dialog.

If you select to have no file associated or to keep the current item, the wizard finishes.

A.76.3 Switch Content Wizard: Check-In Content Screen

In this screen of the Switch Content wizard, you must fill out the form for checking in the content. This screen is no different from the other times you check content in to the server.

After checking in the content to the content server, you click to go to the next screen and then finish the wizard.

A.76.4 Switch Content Wizard: Confirmation Screen

The confirmation screen opens after you have selected the content, or selected to display no content.

If you selected a subtemplate, then the screen displays "Switch Subtemplate Confirmation" with the appropriate content information name listed.

A.77 Switch Region Template Dialog

When the placeholder definition allows switching region templates, the context dialog on the contribution graphic includes "Switch Region Template."

Element	Description
Use the default region template	Select to use the region template marked as the default in the placeholder definition.
Use this region template	Select to open the menu of all region templates.
OK	Select to switch to the selected region template.
Cancel	Cancel your changes and closes the dialog.
Help	Opens the online help for this dialog.

When this dialog is opened in Designer, all region templates associated with the web site are available in the menu, and not just those listed in the placeholder definition. When this dialog is opened in Contributor, only those region templates listed in the placeholder definition are available in the menu.

A.78 Insert Dynamic Conversion Dialog

The Insert Dynamic Conversion dialog is used when you insert a dynamic conversion in a region template.

Element	Description
Name	Use to select the specific dynamic conversion to use for converting native documents.
OK	Saves your settings and closes the Insert Dynamic Conversion dialog.
Cancel	Cancel your settings and closes the Insert Dynamic Conversion dialog.
Help	Opens the online help for this specific dialog.

A.79 Custom Configuration Scripts Dialog

Custom Configuration Scripts are used to modify the customization scripts associated with an element definition.

Element	Description
Custom Configuration Scripts	The name, including path, of the script.

Element	Description
	Creates a script. The path can be typed in completely, or the search button, which then appears, performs a search on the content server.
	Deletes the configuration script.
	Moves a script up in the list. The scripts load in order from top to bottom.
	Moves a script down in the list. The scripts load in order from top to bottom.
OK	Saves your settings and closes the Custom Configuration Script dialog.
Cancel	Cancel your settings and closes the Custom Configuration Script dialog.
Help	Opens the online help for this specific dialog.

A.80 Contribution Element Dialog

The Contribution Element dialog is used to add an element to a region template. If the region template does not have a region definition associated with it, then elements can still be added, however, until the region definition is created, the elements do not function.

Element	Description
Name	The name of the element as it appears in the Designer view. The elements available in the menu are the elements in the region definition.
Region Definition ID	The name of the region definition associated with the region template.
OK	Saves your settings and closes the Define Placeholder dialog.
Cancel	Cancel your settings and closes the Define Placeholder dialog.
Help	Opens the online help for this specific dialog.
Add Element	Opens the Element dialog (see Section A.43, "Element Dialog") to add an element definition to the region definition associated with the region template.

A.81 Select Fragment Dialog

The Fragment Toolbox is only available using the mouse. To allow fragment use with other input devices, you use the Select Fragment dialog.

Element	Description
Available Fragments	The list of fragments broken down by type.
Insert	Inserts the selected fragment on the template.
Edit	Opens the selected fragment in the Fragment Editor dialog (see Section A.57, "Fragment Editor Dialog").

Element	Description
Copy and Edit	Makes a copy of the selected fragment and open the copy in the Properties for Fragment dialog (see Section A.58, "Properties for Fragment Dialog"). After entering new ID information for the copy, the copy of the fragment will then open in the Fragment Editor dialog (see Section A.57, "Fragment Editor Dialog").
Delete	Deletes the selected fragment.
Preview	Opens the Fragment Preview dialog (see Section A.66, "Fragment Preview Dialog"), where you can preview the fragment. You may be prompted to choose a set of parameter values to preview the fragment.
Done	Closes the dialog.

A.82 Oracle Site Studio Replicator

Oracle Site Studio Replicator is a tool that you use with Oracle Site Studio to replicate portions of your site hierarchy from one content server (the *source* server) to another (the *target* server). Without Oracle Site Studio Replicator, you must replicate the entire site hierarchy. Oracle Site Studio Replicator can also be used to override environment properties on your site.

Oracle Site Studio Replicator only replicates your site hierarchy; it does not replicate the content on your site (page templates, data files, fragments, and so on).

This section covers the following topics:

- [Section A.82.1, "Step 1: Select Source Server"](#)
- [Section A.82.2, "Step 2: Select Target Server"](#)
- [Section A.82.3, "Step 3: Select Items to Replicate"](#)
- [Section A.82.4, "Step 4: Perform Replication"](#)
- [Section A.82.5, "Modifying Server Connections"](#)
- [Section A.82.6, "Advanced Options"](#)
- [Section A.82.7, "Override Environment Properties Dialog"](#)

A.82.1 Step 1: Select Source Server

This screen is used to select a source server and web site on the server. This is the server where you are replicating your site hierarchy from. You can select a source server by selecting the actual server or a project file created on the source server. Once you select a server, you then select a web site.

Element	Description
Source Content Server	The name of the Oracle Content Server to replicate a web site from. The Additional Information button opens the Server Configuration dialog (see Section A.82.5, "Modifying Server Connections"), a dialog where you enter the Cgi Url of the Oracle Content Server and a name for the Oracle Content Server.
Site Label	The name of the web site on the Oracle Content Server.
Site ID	The Site ID of the Web Site on the Oracle Content Server.

Element	Description
Back	Takes you to the previous screen.
Next	Takes you to the next screen in the dialog.
Cancel	Cancels your settings and closes the dialog.
Help	Opens the online help for this specific dialog screen.

A.82.2 Step 2: Select Target Server

This screen is used to select a target server. This is the server where you are replicating your site hierarchy to. You can select the server by selecting the actual server or a project file on the target server.

Element	Description
Target Content Server	The name of the Oracle Content Server to replicate a web site to. The Additional Information button opens the Server Configuration dialog (see Section A.82.5, "Modifying Server Connections"), a dialog where you enter the Cgi Url of the Oracle Content Server and a name for the Oracle Content Server.
Advanced	Opens the Advanced dialog, which allows you to chose between replicating the hierarchy directly to the target server, or by creating a hierarchy information file to use to manually replicate the web site.
Back	Takes you to the previous screen.
Next	Takes you to the next screen in the dialog.
Cancel	Cancels your settings and closes the dialog.
Help	Opens the online help for this specific dialog screen.

A.82.3 Step 3: Select Items to Replicate

This screen is used to select individual sections, custom properties, and asset categories to replicate. You can also override any environment properties on the site.

The source hierarchy appears in the Site Hierarchy pane on the left, and the merged hierarchy appears in the Preview pane on the right.

Element	Description
Site Hierarchy	Select the entire web site, or sections of the web site to replicate.
Preview	The hierarchy of the selected objects to replicate.
Advanced	Opens the Advanced dialog (see Section A.82.6, "Advanced Options"). This dialog allows you to determine how much the replication matches the original web site. This includes the switched site asset associations, the environment properties, and how much of the hierarchy is preserved in replication.
Back	Takes you to the previous screen.
Next	Takes you to the next screen in the dialog.
Cancel	Cancels your settings and closes the dialog.
Help	Opens the online help for this specific dialog screen.

A.82.4 Step 4: Perform Replication

The Perform Replication screen is used to display the location and URL of both the source and target servers, the replication type, and the site hierarchies to be replicated.

Element	Description
Preview	Opens a display to show the hierarchy (or the selected portion of the hierarchy) of the replicated, merged web site.
Info	Displays an overview of the Source and Target servers, as well as the status of the hierarchy to be replicated.
Back	Takes you to the previous screen.
Finish	Starts replication with the entered parameters.
Cancel	Cancels your settings and closes the dialog.
Help	Opens the online help for this specific dialog screen.

A.82.5 Modifying Server Connections

This dialog is used to add or modify a server as a source or target location. You can change the properties of an existing server, add a server, and delete a server.

Element	Description
Server Name	Enter the name of the Oracle Content Server you will use.
CGI URL	Enter the Cgi Url of the Oracle Content Server to use.
OK	Stores the Oracle Content Server configuration information, and closes the Server Configuration dialog.
Cancel	Cancels your settings and closes the dialog.
Help	Opens the online help for this specific dialog screen.

A.82.6 Advanced Options

This dialog is used to modify what specific relationships of the hierarchy and site asset structure in the source are carried over to the target server.

Element	Description
Options	Select to either replicate only the folders selected on the Select Items to Replicate screen (see Section A.82.3, "Step 3: Select Items to Replicate") or to replicate all folders in the hierarchy.
Override Environment Properties	Opens the Override Environment Properties dialog (see Section A.82.7, "Override Environment Properties Dialog").
Preserve the target's switched region associations	Select to preserve the current switched region associations in the site.
Preserve the target's switched placeholder definition associations	Select to preserve the current switched placeholder definition associations in the site.
Preserve the target's switched region template and subtemplate associations	Select to preserve the current switched region template and subtemplate associations in the site.
OK	Stores the options and closes the Advanced Options dialog.

Element	Description
Cancel	Cancels your settings and closes the dialog.
Help	Opens the online help for this specific dialog screen.

A.82.7 Override Environment Properties Dialog

When replicating a web site, the properties of the site are replicated as well from one server to the next. In this dialog, you can specify properties that are not replicated and instead remain on the source server. The settings are called environment properties.

Element	Description
Name	The name of the property. Check the box beside the name to override the current environment property settings.
Type	The type of property.
Description	A description of the property.
OK	Saves your settings and closes the Override Environment Properties dialog and returns you to the Set Items to Replicate dialog (see Section A.82.3, "Step 3: Select Items to Replicate").
Cancel	Cancels your settings and closes the Override Environment Properties dialog and returns you to the Set Items to Replicate dialog (see Section A.82.3, "Step 3: Select Items to Replicate").
Help	Opens the online help for this specific dialog.

Replicating Your Site

The web sites that you create with Oracle Site Studio reside on a single instance of the content server. At some point, you may want to copy the site from one content server to another. This process is called *replication*, and it is often used by administrators to reproduce a site on two (or more) content servers. For example, a development server and a production server. The copying process can be automated so that your site remains current on all servers.

This section covers the following topics:

- [Section B.1, "About Replication"](#)
- [Section B.2, "Configuring the Target Server for Replication"](#)
- [Section B.3, "Setting Up an Outgoing Provider on the Source Server"](#)
- [Section B.4, "Creating the Import and Export Archives"](#)
- [Section B.5, "Starting the Site Replication"](#)
- [Section B.6, "Specifying Environment Properties"](#)
- [Section B.7, "Replication Individual Sections of Your Site"](#)
- [Section B.8, "Using Oracle Site Studio Replicator"](#)

B.1 About Replication

To better understand replication, you should be aware of the terminology and processes that are involved. Because Oracle Site Studio relies on the content server's Archiver/Replicator tool to replicate the site, you can refer to the content server documentation for more information.

To replicate your site, you must perform several steps to ensure that the site works properly on the server you're copying it to (the target server). You must prepare that server environment, configure the two servers so that they communicate with one another, set up archives on both servers, and perform the replication.

You find most of the replication features on the Manage Site Replication page in the content server, but you can also use Oracle Site Studio Replicator to customize the replication even more. You can use this four-part wizard to replicate individual sections of the site, merge sections on the target server, and override environment properties.

When you replicate a site, you copy the site hierarchy, site assets, and more. The replication feature in Oracle Site Studio uses the existing replication framework in the content server (Archiver/Replicator). As such, if you've worked with that tool before, many of the steps in Oracle Site Studio will already be familiar to you.

Replication is described in more detail in these sections:

- [Section B.1.1, "The Replication Process"](#)
- [Section B.1.2, "What Gets Replicated"](#)
- [Section B.1.3, "Revisions Included"](#)
- [Section B.1.4, "Additional Export Settings"](#)

B.1.1 The Replication Process

The replication process in Oracle Site Studio includes a source server, a target server, an export archive, an import archive, and the transfer.

Figure B-1 The Stages Involved in Replication

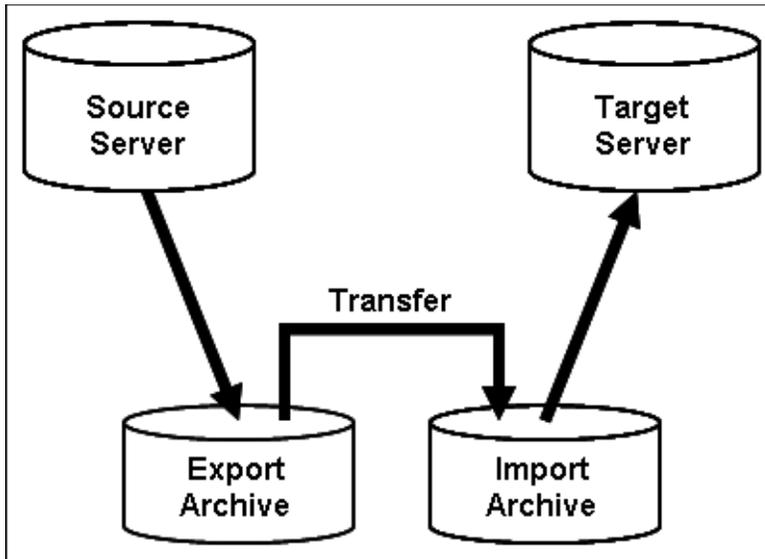


Diagram of the replication process, which is described in the surrounding text.

- **Export:** Native files, web-viewable files, content types, and user attributes are copied out of the source content server into an export archive.
- **Import:** Files and content server information are copied from an import archive to the target content server.
- **Transfer:** Content is transferred from one content server to another. This function can be used to copy content across a firewall or between two servers that do not have access to the same file system. You can also transfer archives between two content servers that have access to the same file system.
- **Replicate:** Automates the export, import, and transfer steps. You can use replication to automatically export from one content server, transfer the archive to another computer, and then import to another content server.

The replication process consists of several steps:

1. [Section B.2, "Configuring the Target Server for Replication"](#)
2. [Section B.3, "Setting Up an Outgoing Provider on the Source Server"](#)
3. [Section B.4, "Creating the Import and Export Archives"](#)

4. [Section B.5, "Starting the Site Replication"](#)
5. [Section B.6, "Specifying Environment Properties"](#)

B.1.2 What Gets Replicated

When you use the Oracle Site Studio Replication feature to replicate your site from one content server to another, the following items are included:

- **Project file** (the dDocName of the project file)
- **Web site ID** (where the xWebsites field contains the SiteID of the site)
- **Web site sections** (where the xWebsiteSection field contains the SiteID of the site)
- **Fragment libraries** (where the xWebsiteObjectType field contains the value "fragment")

It is important that the metadata values for xWebsites, xWebsiteSection, and xWebsiteObjectType be used correctly on your site in order for the contents of the site to be replicated.

B.1.3 Revisions Included

Revisions with a release date later than the most recent export date are exported. This prevents archives from exporting material that has already been exported, which ensures the migration of content only one time and prevents the unfettered growth in the size of the archives. Furthermore, all selected revisions of content that match the export query are exported to the archive. This works in tandem with the release date filter to ensure that the required web site content is replicated.

B.1.4 Additional Export Settings

The Oracle Site Studio Replication feature includes several other useful settings to ensure that your site is successfully replicated:

- Configuration information for the content is included for administrators who plan to synchronize metadata models between two content servers. (Oracle Site Studio does not use this information.)
- The source archive is identified as the transfer owner (the content server that initiates the transfer from the source archive to the target archive).
- The import archive on the target server is identified as the target archive, which takes responsibility for receiving material exported and transferred from a source server and then imports the material into the target server.

B.2 Configuring the Target Server for Replication

The Oracle Site Studio replication feature copies your site from one content server to another. It does not, however, copy the content server environment from the source server to the target server.

Before you replicate the site, you must make sure the content server environment on the target server is set up the same way as the source server (at least the parts used by your site). If you skip this step, your web site experiences problems during the replication process.

The steps to reproduce the content server environment varies from one organization to another and from one web site to another. As such, we offer these general guidelines.

To manually configure the target server environment, perform these tasks:

1. Recreate the metadata model used on the source server. This includes all custom metadata fields and new file formats. This metadata is required for your site to operate properly. For example, checking in files generally relies on your metadata settings, and certain features on your site rely on custom metadata fields.
2. Reinstall all components used by your site on the source server. This includes Dynamic Converter (for native documents) and Check Out and Open (to check native documents out using the contribution icon).
3. Recreate unmanaged resources. This includes custom ActiveX controls or JSP objects that are used by your site.
4. Recreate any additional configuration settings that you have introduced to the server. This includes anything that has changed the behavior of the server.

If you run version 7.5.2 or later of Oracle Content Server (including 10gR3), then you may be able to use the Content Server Configuration Migration Utility to replicate the content server environment. The next step is to set up an outgoing provider (see [Section B.3, "Setting Up an Outgoing Provider on the Source Server"](#)).

B.3 Setting Up an Outgoing Provider on the Source Server

In Oracle Content Server, a provider is an API (Application Programming Interface) that establishes a connection between two or more content servers. To replicate a web site, you must create an outgoing provider on the source server that establishes a connection between it and the target server.

After you configure the target server (see [Section B.2, "Configuring the Target Server for Replication"](#)), you are ready to set up an outgoing provider on the source server.

To set up an outgoing provider on the source server, perform these tasks:

1. Log on to the source content server as an administrator. On the main menu, select **Administration** and then **Providers**.

Figure B–2 Provider Page in Oracle Content Server User Interface

Provider	Type	Connection State	Last Activity Date	Actions
SystemDatabase System Database	database	5 out of 5 connections are good	1/8/04 8:49 AM	Info Test
SystemServerSocket System Server Socket	incoming	good	1/8/04 8:49 AM	Info Test

Provider Type	Description	Action
outgoing	Configuring an outgoing provider.	Add
database	Configuring a database provider.	Add
incoming	Configuring an incoming provider.	Add
preview	Configuring a preview provider.	Add
ldapuser	Configuring an LDAP user provider.	Add

2. Under Create a New Provider, click **Add** beside the *outgoing* provider type.

This opens the Add Outgoing Provider page.

Figure B-3 Add Outgoing Provider Page

Add Outgoing Provider	
Provider Name	<input type="text" value="Archiver_Provider"/>
Provider Description	<input type="text" value="Provider to web sites"/>
Provider Class	<input type="text" value="intradoc.provider.SocketOutgoingProvider"/>
Connection Class	<input type="text" value="intradoc.provider.SocketOutgoingConnection"/>
Configuration Class	<input type="text"/>
Server Host Name	<input type="text" value="TargetServer"/>
HTTP Server Address	<input type="text"/>
Server Port	<input type="text" value="4444"/>
Instance Name	<input type="text" value="idc_production_server"/>
Relative Web Root	<input type="text" value="/stellent/"/>
Server Options:	<input type="checkbox"/> Proxied <small>Web access and security of a remote server is controlled by this server. Only enable this option if you are the master server in a master and proxied server relationship. Do not enable this option if you only wish to transfer archives.</small> <input type="checkbox"/> Notify Target <small>Use this option if you are the proxied server in a master and proxied server relationship. The <code>/usrz</code> subject gives the master server's web server access to the security configuration of this server and guarantees that its copy is kept up to date. It should be checked if you wish static content on the proxied server to be directly available through the master server's web server. The <code>Released Documents</code> subject should be checked if you wish to perform an enterprise search from the master server which includes this proxied server.</small> <input type="checkbox"/> Users <input type="checkbox"/> Released Documents

3. For Provider Name, enter a name to identify the provider (see illustration above). The name appears in the Providers list on this page (after you add one), and it becomes a subdirectory in the `[CS-Dir]/data/providers/` on the source server.
4. For Provider Description, enter a friendly description for the provider (see illustration above).
5. For Provider Class, enter **intradoc.provider.SocketOutgoingProvider** (the name of the Java class for outgoing providers).
6. For Connection Class, enter **intradoc.provider.SocketOutgoingConnection** (the name of the Java class that implements provider connections).
7. You can leave Configuration Class empty (this identifies a Java class for additional configuration settings, useful for database providers).
8. For Server Host Name, enter the name (typically, the system name or IP address) of the target server.
A socket connection is made to this host.
9. You can leave HTTP Server Address empty (this is the HTTP address of the target server, which is not necessary for this type of connection).
10. For Server Port, enter the port (typically, 4444) used to communicate with the target server.

You can determine the port by viewing the server output when starting the content server.

11. For Instance Name, enter the name (IDC_Name) of the target content server.
12. For Relative Web Root, enter the relative web root for the target content server (for example, */stellent/*).

You can skip the remaining options as they are not required for this type of connection.

13. Press **Add** to save the provider information and return to the Providers page. (Your outgoing provider is in the Providers list.)
14. Restart the content server.

To test the provider and make sure it was properly set up, return to the **Providers** page and click **Test** beside the outgoing provider.

In addition to setting up the outgoing provider, you may still must configure the server IP address filter on the target system so that the source server is allowed to communicate with the target server (for more information, see the Oracle Content Server Help).

If there is a firewall in place, you must allow connections from the source server to the target server on the port defined in step 10.

The next step is to create an import archive (see [Section B.4.1, "Setting Up the Import Archive on the Target Server"](#)).

B.4 Creating the Import and Export Archives

When you replicate your site from one content server to another, the Oracle Site Studio replication feature exports the contents of your site into an export archive. The export archive content is copied into an import archive, residing on the target server. The import archive is then extracted to that server so that your site can be viewed and used again on the new server.

You create the export archive on the source server and an import archive on the target server. You must create the import archive on the target server first, however, so that you can point the export archive to that location when you set one up.

- [Section B.4.1, "Setting Up the Import Archive on the Target Server"](#)
- [Section B.4.2, "Setting Up the Export Archive on the Source Server"](#)

B.4.1 Setting Up the Import Archive on the Target Server

An import archive resides on the target content server. During replication, content from an export archive (residing on a source server) is copied into the import archive. The import archive then copies the content to the target content server, thus completing the replication process.

After you set up a provider connection (see [Section B.3, "Setting Up an Outgoing Provider on the Source Server"](#)), you're ready to create an import archive. You create (or edit) an import archive on the Manage Site Replication page on the target server.

To set up an import archive, perform these tasks:

1. Log on to the target content server as an administrator. On the main menu, select **Administration**, then **Oracle Site Studio Administration**, and then **Manage Site Replication**.

This opens the Manager Site Replication page.

2. Click Add Import Archive.

This opens the Add Import Archive page.

Or, if you are updating an existing import archive, highlight it in the list of archives and click **Change Settings**.

3. Enter a name for the archive beside Archive Name (the name appears in the list of available archives on the Manage Site Replication page).

The archive name should not contain spaces or special characters.

4. Enter a description of the archive beside Archive Description.

5. To retain content that contributors assign to each region on the target site, check Retain switched region content on target server during import.

These are the areas where a contributor has switched the data file or native document.

6. To retain content that contributors edit in each region on the target site, check Retain region content on target server during import.

This prevents any data files and native documents from being copied from the source server to the target server and potentially overwriting files edited by the contributor.

7. Click Add Archive.

Or, if you are updating an existing import archive, click **Update**.

You would typically choose to retain region content and switched region content if you're replicating from a development server (source) to a contribution server (target) because in that scenario you want to preserve the changes made by a contributor.

You would likely disable (uncheck) these options if you're replicating from a contribution server (source) to a consumption server (target) because in that scenario you want to overwrite anything that has changed on the consumption server.

The next step is to create an export archive (see [Section B.4.2, "Setting Up the Export Archive on the Source Server"](#)).

B.4.2 Setting Up the Export Archive on the Source Server

An export archive resides on the source content server, where it collects information from the web site. During replication, the export archive is copied to an import archive, residing on the target server. (That import archive must be created first so that you can point to it when you create the export archive.)

After you create an import archive (see [Section B.4.1, "Setting Up the Import Archive on the Target Server"](#)), you're ready to create an export archive. You create (or edit) an export archive using the Manage Site Replication page on the source server.

To set up an export archive, perform these tasks:

1. Log on to the source content server as an administrator. On the main menu, select Administration, then Oracle Site Studio Administration, and then Manage Site Replication.

This opens the Manage Site Replication page.

2. Click Add Export Archive.

This opens the Add Export Archive page.

Or, if you are updating an existing export archive, highlight it in the list of archives and click **Change Settings**.

3. Enter a name for the archive beside Archive Name (the name appears in the list of available archives on the Manage Site Replication page).

The archive name should not contain spaces or special characters.

4. Enter a description of the archive beside Archive Description.
5. Choose the web site to replicate in the **Web Site** menu.
6. Check **Include project file in export archive** to replicate your entire site hierarchy.

If you do not want to replicate the entire site hierarchy and instead want to replicate individual sections, then leave this option unchecked, continue with the steps below and then follow the steps in [Section B.7, "Replication Individual Sections of Your Site."](#)

7. For Transfer to Collection, select the collection on the target content server where this archive is copied to.

Click **Open Collection** to browse to the collection on the target server.

8. For Transfer to Archive, select the archive on the target server where this export archive is copied to (see [Section B.4.1, "Setting Up the Import Archive on the Target Server"](#)).
9. Check **Automatically export new and changed content** if you want the replication process to be automatic whenever content on the source server changes. (If you do not choose this, then you must manually trigger the replication.)
10. Click **Add Archive**.

Or, if you are updating an existing export archive, click **Update**.

When you select a web site for the export archive (Step 7), Oracle Site Studio archives and replicates everything related to your site (see [Section B.1.2, "What Gets Replicated"](#)). To customize what gets archived or to add more items to the archive, you can refine the archive query directly using the Archiver applet.

The next step is to start the replication (see [Section B.5, "Starting the Site Replication"](#)).

If you plan to use Oracle Site Studio Replicator to replicate individual sections of your site, then the next step is opening the Oracle Site Studio Replicator (see [Section B.8, "Using Oracle Site Studio Replicator"](#)).

B.5 Starting the Site Replication

After you've set up an export archive (see [Section B.4.2, "Setting Up the Export Archive on the Source Server"](#)), you're ready to initiate the replication process on the source server using the Manage Site Replication page.

To replicate your web site, perform these tasks:

1. Log on to the source content server as an administrator. On the main menu, select **Administration**, then **Oracle Site Studio Administration**, and then **Manage Site Replication**.

This opens the Manage Site Replication page.

2. Select the desired export archive.

3. Click **Export**.

If you chose to automatically export new and changed content (Step 10 in [Section B.4.2, "Setting Up the Export Archive on the Source Server"](#)), then the replication process is automatic after you export the archive.

Note: The first time you export a web site, it must be done manually. If the site is set for automatic export, de-select the **Automatically export new and changed content** checkbox, then click **Export** to run the first export manually. After this first manual export, replication can be made automatic.

Depending on the size and complexity of your site, you may must allow plenty of time for the target content server to re-index your site when the replication is complete.

B.6 Specifying Environment Properties

When replicating your site from one location to another, you may want to preserve certain settings from one server and not replicate them to the next server. These settings are called environment properties.

If, for example, you want to specify a unique caching value on one server (a contribution site) and not the next server (a customer site), you would specify that the max-age property be an environment property. As a result, it is not replicated to the next server.

You specify environment properties in Designer, and you can override those values in Replicator.

To specify environment properties, perform these tasks:

1. With your site open in Designer, click **File** then **Site**, then select **Advanced** then **Define Environment Properties**.

This opens the Define Environment Properties dialog (see [Section A.7, "Define Environment Properties Dialog"](#)).

2. Check the box beside the property to be an environment property (and therefore not replicated).
3. Clear the box beside the property to be treated normally and therefore replicated to the next location (these properties are called application properties).
4. Click **OK**.

You can also override environment properties during replication (see [Section B.8.6, "Overriding Environment Properties"](#)).

B.7 Replication Individual Sections of Your Site

Instead of replicating an entire web site in one action, you may want to replicate individual sections of the site. This is useful, for example, if you have a section that is not ready for public viewing, so you hold off on replicating that section until it is ready.

Or perhaps, you have a section that is only intended for a production environment and not a consumption environment. As such, you would want to omit that section when replicating the site hierarchy.

You use Oracle Site Studio Replicator to replicate individual sections of your site. For more information, see [Section B.8, "Using Oracle Site Studio Replicator."](#)

Before you proceed, you should verify that you have unchecked the project file in your export archive (Step 7 in [Section B.4.2, "Setting Up the Export Archive on the Source Server"](#)). If not, your replication settings in Oracle Site Studio Replicator are overwritten by the replication of the entire site hierarchy.

B.8 Using Oracle Site Studio Replicator

Oracle Site Studio Replicator is a tool that you use with Oracle Site Studio to replicate portions of your site hierarchy from one content server (the *source* server) to another (the *target* server). Without Oracle Site Studio Replicator, you must replicate the entire site hierarchy. Oracle Site Studio Replicator can also be used to override environment properties on your site.

Oracle Site Studio Replicator only replicates your site hierarchy; it does not replicate the content on your site (page templates, data files, fragments, and so on). This content is replicated when you use the Manage Site Replication page.

Starting Oracle Site Studio Replicator

Oracle Site Studio Replicator is installed locally on your system with Designer when you install Oracle Site Studio. You can launch this program from your Start menu: select **Programs**, then select **Oracle Universal Content Management**, then select **Oracle Site Studio 11gR1**, and choose **Oracle Site Studio Replicator**.

With the program open, you are ready to go through the wizard and replicate your site hierarchy. The replication process consists of several steps:

- [Section B.8.1, "Step 1: Selecting a Source Server and Web Site"](#)
- [Section B.8.2, "Step 2: Selecting a Target Server"](#)
- [Section B.8.3, "Step 3: Selecting Items to Replicate"](#)
- [Section B.8.4, "Step 4: Completing Oracle Site Studio Replicator"](#)

B.8.1 Step 1: Selecting a Source Server and Web Site

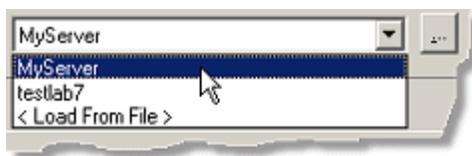
The first screen in Oracle Site Studio Replicator asks you to select a source server. This is the server where you are replicating your site hierarchy from. You can select a source server by selecting the actual server or a project file created on the source server. Once you select a server, you then select a web site. For more information, see [Section A.82.1, "Step 1: Select Source Server."](#)

Selecting a Server as the Source

To select a server as the source, perform these tasks:

1. Select a server from the Source Content Server drop-down list.

Figure B-4 Selecting a Server



To add a server, edit an existing one, or remove one, click the **Additional Information** icon (Figure B-5) beside the drop-down list (see Section B.8.5, "Modifying Server Connections").

Figure B-5 Additional Information Icon



2. Click **Next** (and when prompted, log onto the content server).
3. Select a web site on the source server.

Figure B-6 Selecting a Web Site

Site Label	Site ID
appFolder	appFolder
B/C's bug bashing site	bc_bug_bash
bin_asp	bin_asp
bin1	bin1
bul01	bul01001
bul01_07	bul011_07
MyWebSite	MyWebSite
NewSite	NewSite
Rowena's Hosting Tutorial	9001
rtb	rtb

4. Click **Next** and proceed to the next step (see Section B.8.2, "Step 2: Selecting a Target Server").

Selecting a Project File as the Source

You can use a project file as the source, but you must first download the project file from the source server (see Section 7.3.4, "Viewing the Content Information Page for the Project File"). Using a project file instead of a source server is useful when you must manually transfer the hierarchy because the source server is not accessible on the network.

To select a project file as the source, perform these tasks:

1. Select **<Load From File>** from the Source Content Server drop-down list.
2. Click **Next**.
3. From the Open dialog, select the project file that you downloaded from the source server and click **OK**.
4. Click **Next** and proceed to the next step (see Section B.8.2, "Step 2: Selecting a Target Server").

Later in the wizard, you have the opportunity to select individual sections from the web site you are replicating.

B.8.2 Step 2: Selecting a Target Server

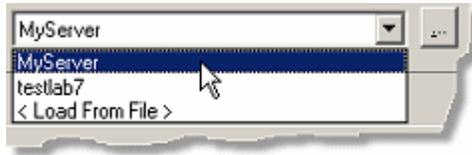
The next screen in Oracle Site Studio Replicator asks you to select a target server. This is the server where you are replicating your site hierarchy to. You can select the server by selecting the actual server or a project file on the target server. For more information, see Section A.82.2, "Step 2: Select Target Server."

Selecting a Server as the Target

To select a server as the target, perform these tasks:

1. Select a server from the Target Content Server drop-down list. (It is not valid to leave the Target Content Server field blank.)

Figure B-7 *Selecting a Server*



To add a server, edit an existing one, or remove one, click the **Additional Information** icon (Figure B-8) beside the list of servers (see Section B.8.5, "Modifying Server Connections").

Figure B-8 *Additional Information icon*



2. Click **Advanced** and choose an option:
 - Choose **Replicate hierarchy information directly to the target content server** to replicate to the target content server directly.
This option is only available when you choose a server as the target (as opposed to a project file).
 - Choose **Generate a hierarchy information file for manual replication** to generate a hierarchy information file.
Specify an **Output File Name** for the hierarchy information file or click **Browse** to locate one.
By choosing this option, a project file is created that you can later check into the content server.
3. Click **OK** to return to Oracle Site Studio Replicator.
4. Click **Next** (and log onto the content server, if prompted).
5. If prompted to create a web site (the project file), click **Yes**.
6. Proceed to the next step (see Section B.8.3, "Step 3: Selecting Items to Replicate").

Selecting a Project File as the Target

You can use a project file as the target, but you must first download the project file from the target server (see Section 7.3.4, "Viewing the Content Information Page for the Project File"). Using a project file instead of a target server is useful when you must manually transfer the hierarchy because the target server is not accessible on the network.

To select a project file as the target, perform these tasks:

1. Select **<Load From File>** from the Target Content Server drop-down list.
2. Click **Next**.
3. Click **OK** when prompted that an output file name is required.
4. In the **Output File Name** field, specify a name for the hierarchy information file.
Or, click **Browse** to locate an existing file.

5. Click **OK** to return to Oracle Site Studio Replicator.
6. Click **Next**.
7. In the Open dialog, select the hierarchy information file to use and click **Open**.
8. Proceed to the next step (see [Section B.8.3, "Step 3: Selecting Items to Replicate"](#)).

B.8.3 Step 3: Selecting Items to Replicate

After you select your source and target servers (or project files), the next screen opens, where you can select individual sections, custom properties, and asset categories to replicate. You can also override any environment properties on the site.

The source hierarchy appears in the Site Hierarchy pane on the left, and the merged hierarchy appears in the Preview pane on the right. For more information, see [Section A.82.3, "Step 3: Select Items to Replicate."](#)

To select items to replicate, perform these tasks:

1. In the Site Hierarchy pane, check the boxes beside the desired sections, custom properties, and asset categories you want to replicate.

The first time you replicate a site, you see the complete site hierarchy (including the root) in the Preview pane, even though nothing has been selected yet on the source. You still must select the site hierarchy and root section on the source in order for the site hierarchy to replicate.

2. Click **Advanced** and choose an option for replicating your hierarchy:

- Choose **Replicate the selected folders** to replicate individual sections of your site.
- Choose **Replicate the entire source hierarchy** to replicate the entire site.

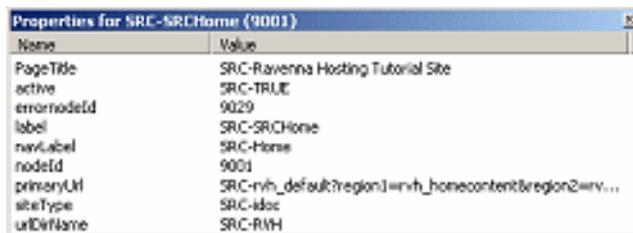
Note: Replicating the entire hierarchy is common if this is the first time you are replicating the site. The next time you replicate, you would choose selected folders.

- Click **Override Environment Properties** to override the environment settings you made in the Environment Properties dialog in Designer (see [Section B.8.6, "Overriding Environment Properties"](#)).
- Check **Preserve the target's switched region associations** to preserve the content that contributors switched on the target server and therefore prevent it from being overwritten.

Click **OK** to return to Oracle Site Studio Replicator.

3. To view the current properties for a particular section (on the source or target server), right-click that section and choose **Properties**.

This opens the Properties dialog.

Figure B–9 Properties Dialog


Name	Value
PageTitle	SRC-Ravenna Hosting Tutorial Site
active	SRC-TRUE
errornodeId	9029
label	SRC-SRCHome
navLabel	SRC-Home
nodeId	9001
primaryUrl	SRC-rvh_default?region1=rvh_homecontent®ion2=rv...
siteType	SRC-idoc
urlDirName	SRC-RVH

With this window open, you can continue clicking on different sections of the site hierarchy to view the properties for that section.

4. To remove a section from the target server, right-click that section and choose **Delete**.

Deleting a section from the target server is useful when a section is no longer needed, or it has been moved to another part of the site.

5. Click **Next** and proceed to the next step (see [Section B.8.4, "Step 4: Completing Oracle Site Studio Replicator"](#)).

You can resize this dialog to see more of your site hierarchy.

When you select a web site at the top-level or a section at the top-level, every section below it is replicated.

If you select a subsection on the source server, then the parent of that subsection must already exist on the target server. If not, you receive a message stating that the selected hierarchies could not be merged because there is no parent in the target hierarchy. To resolve, you must select the parent section of the one you're replicating on the source hierarchy.

A section that appears in bold on the target hierarchy (Preview pane) indicates that it is a section that is created during the replication process.

B.8.4 Step 4: Completing Oracle Site Studio Replicator

After you select your source and target servers (or project files) and have selected items to replicate, the next screen opens, where you can confirm your choices and replicate the site hierarchy.

On this screen, you see the location and URL of both the source and target servers, the replication type, and the site hierarchies to be replicated. For more information, see [Section A.82.4, "Step 4: Perform Replication."](#)

To replicate the site hierarchy, perform these tasks:

1. Click **Preview** to see the site hierarchy changes that takes place on the target server. When finished, click **Close**.
2. Click **Finish**.
3. Click **Yes** to the message asking to replicate to the target server.
4. If you are creating a site on the target server, you are taken to the **Assign Info Form** where you specify the metadata that is assigned to the project file.

When finished, click **Assign Info**.

5. Click **OK** to the confirmation message.

If this is the first time you are replicating your site, then return to the Manage Site Replication page to complete the replication (see [Section B.5, "Starting the Site Replication"](#)).

B.8.5 Modifying Server Connections

When you select a server as a source or target location, you can change the properties of an existing server, add a server, and delete a server.

Adding a New Server to the List

To add a server to the list, perform these tasks:

1. When specifying a source or target server in Oracle Site Studio Replicator, click the **Additional Information** icon ([Figure B-10](#)).

Figure B-10 Additional Information Icon



2. Click **Add Server**.
3. In the Server Configuration dialog, enter the name for the server in the **Server Name** field (this is what opens the next time you choose a server).
4. Enter the URL of the server in the **CGI URL** field.
Typically, you can use the existing URL provided in the field, replacing *server* with the name of your server.
5. Click **OK**.

Removing a Server From the List

To remove a server from the server list, perform these tasks:

1. When specifying a source or target server in Oracle Site Studio Replicator, select the server you want to remove, and then click the **Additional Information** icon ([Figure B-11](#)).

Figure B-11 Additional Information Icon



2. Click **Remove Server**.
3. Click **Yes** to the confirmation message asking you to remove the server from the list.

Viewing or Editing a Server's Properties

To view or edit a server's properties, perform these tasks:

1. When specifying a source or target server in Oracle Site Studio Replicator, select the server you want to view or edit, and then click the **Additional Information** icon ([Figure B-12](#)).

Figure B-12 Additional Information Icon



2. Click **Properties**.
3. In the Server Configuration dialog, you can view or edit the properties of the server.
4. Click **OK**.

B.8.6 Overriding Environment Properties

After you've specified environment properties for your site in Designer (see [Section B.6, "Specifying Environment Properties"](#)), you can later override those settings when you replicate the site.

You do this in the Override Environment Properties dialog, which is located in Oracle Site Studio Replicator (see [Section B.8.3, "Step 3: Selecting Items to Replicate"](#)).

To override environment properties, perform these tasks:

1. On the Selecting Items to Replicate screen in Oracle Site Studio Replicator, click **Advanced**.
2. In the Advanced dialog, click **Override Environment Properties**.
3. Check the box beside the property to treat as an environment property.
Or, clear the box beside the property to treat it as an application property.
4. Click **OK** to return to Oracle Site Studio Replicator and complete the replication.

Sample Fragments

Oracle Site Studio includes numerous sample fragments that you can begin using right away without knowing the code that was used to create them. You can use the sample fragments as they are, edit their source code, or create fragments based on existing ones. Starting off with the sample fragments is a great way to learn the structure and the practical use of a fragment.

Many of the sample fragments include parameter that you can use to customize each fragment when you add it to your page template.

The sample fragments are grouped into four categories in the Toolbox. A description of each fragment, along with its available parameters, follows:

- [Section C.1, "Navigation Fragments"](#)
- [Section C.2, "Dynamic List Fragments"](#)
- [Section C.3, "Static List Fragments"](#)
- [Section C.4, "Other Fragments"](#)

C.1 Navigation Fragments

These are the navigation-related fragments:

- [Section C.1.1, "Breadcrumb Plain"](#)
- [Section C.1.2, "Explorer Menu Bar"](#)
- [Section C.1.3, "Login Example"](#)
- [Section C.1.4, "Nav Home Logo"](#)
- [Section C.1.5, "Nav Multi Horizontal"](#)
- [Section C.1.6, "Nav Multi Vertical"](#)
- [Section C.1.7, "Nav Plain Horizontal"](#)
- [Section C.1.8, "Nav Plain Vertical"](#)
- [Section C.1.9, "Nav Tabs Top"](#)
- [Section C.1.10, "Nav Wrap Horizontal"](#)
- [Section C.1.11, "Search Box Plain"](#)
- [Section C.1.12, "Search Result Paged"](#)
- [Section C.1.13, "Search Result Plain"](#)
- [Section C.1.14, "Site Map Plain"](#)

- [Section C.1.15, "Site Map Tree"](#)
- [Section C.1.16, "Vertical Menu Bar"](#)
- [Section C.1.17, "ASP Bread Crumb Plain"](#)
- [Section C.1.18, "ASP Bread Crumb Plain \(server\)"](#)
- [Section C.1.19, "ASP Explorer Menu Bar"](#)
- [Section C.1.20, "ASP Nav Multi Horizontal"](#)
- [Section C.1.21, "ASP Nav Multi Vertical"](#)
- [Section C.1.22, "ASP Search Box Plain"](#)
- [Section C.1.23, "ASP Search Box Plain \(server\)"](#)
- [Section C.1.24, "ASP Search Result Plain"](#)
- [Section C.1.25, "ASP Site Map Plain"](#)
- [Section C.1.26, "CSP Sample Navigation \(client\)"](#)
- [Section C.1.27, "CSP Sample Navigation \(server\)"](#)
- [Section C.1.28, "JSP Nav Home Logo"](#)
- [Section C.1.29, "JSP Nav Plain Horizontal"](#)
- [Section C.1.30, "JSP Nav Plain Vertical"](#)
- [Section C.1.31, "JSP Search Box Plain"](#)
- [Section C.1.32, "JSP Search Results Paged"](#)
- [Section C.1.33, "JSP Site Map Plain"](#)
- [Section C.1.34, "IDOC Breadcrumb Plain"](#)
- [Section C.1.35, "IDOC Nav Home Logo"](#)
- [Section C.1.36, "IDOC Search Box Plain"](#)
- [Section C.1.37, "RSET Breadcrumb Plain"](#)
- [Section C.1.38, "RSET Nav Multi Horizontal"](#)
- [Section C.1.39, "RSET Nav Plain Horizontal"](#)
- [Section C.1.40, "RSET Site Map Plain"](#)
- [Section C.1.41, "Additional IDOC Fragments"](#)

C.1.1 Breadcrumb Plain

Adds navigational links that represent a hierarchical path, also called a breadcrumb. Breadcrumbs are frequently placed at the top or the bottom of a web page, and they help users orient themselves on a web site.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssClassName: Specifies a CSS class name.

ssSeparator: Text used to separate the links. The default is ">".

C.1.2 Explorer Menu Bar

Adds a navigation menu that emulates a hierarchical menu bar, similar to that of a Windows-based application. (This fragment uses DHTML.)

Parameters

ssShowHome: Boolean flag that specifies whether to display a link to the home page. The default is "true."

C.1.3 Login Example

Adds a web site login button and a login script that illustrates how you can take users from a public site to a secure site based on their login credentials.

Parameters

ssButtonText: The text for the "login" button. The default is "Login."

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: CSS Class

ssTargetSiteId: The SiteID of the target (secure) site.

C.1.4 Nav Home Logo

Adds an image that serves as a hyperlink to the home page of the web site.

Parameters

ssLogo: Specifies the URL of the graphic to use in the content server. If this setting is left blank, a placeholder image called "logo.gif" from the fragment assets directory is used.

ssWidth: The width of the image (in pixels). If left blank, no width attribute is used.

ssHeight: The height of the image (in pixels). If left blank, no height attribute is used.

C.1.5 Nav Multi Horizontal

Adds a top navigation bar that can be placed anywhere on the page. This fragment can display a customized range of levels.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

ssSeparator: Text used to separate the parts of the bar. Default is "|".

ssShowHome: Boolean flag that specifies whether to display a link to the home page as part of the top level navigation. The default is "true." (The home page is actually located in the root of the web site, essentially above the "top" level, but it is perceived as a sibling.)

ssStartLevel: The display level to start displaying the menu.

ssNumLevels: The number of (section) levels to display.

Special Instruction

The Nav Plain Horizontal fragment offers a simpler version of this menu.

C.1.6 Nav Multi Vertical

Adds a side navigation bar that can go anywhere on the page template, but is generally placed on the left side where it indents from left to right. This fragment can display a customized range of levels (unlike the Nav Plain Vertical fragment, which opens only the second-level). It opens at the current focus level, so if your site does not have the sub-level sections that you specify in this fragment, then nothing opens.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

ssShowHome: Boolean flag that specifies whether to display a link to the home page as part of the top-level navigation. The default is "false."

ssStartLevel: The display level to start displaying the menu.

ssNumLevels: The number of (section) levels to display.

Special Instruction

The Nav Plain Vertical fragment offers a simpler version of this menu.

C.1.7 Nav Plain Horizontal

Adds a top navigation bar that can be placed anywhere on the page. This fragment opens only the top level of your site navigation; that is, the sections in the root of the site.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

ssSeparator: Text used to separate the parts of the bar. Default is "|".

ssShowHome: Boolean flag that specifies whether to display a link to the home page. The default is "true." (The home page is actually located in the root of the web site, above the "top" level, but it may be perceived as a sibling here.)

Special Instruction

The Nav Multi Horizontal fragment offers a more complex version of this menu.

C.1.8 Nav Plain Vertical

Adds a side navigation bar that can go anywhere on the page template, but is generally placed on the left side where it indents from left to right. This fragment

opens only the second level of the site navigation, so if your site has no second-level sections, nothing opens. The site hierarchy opens in its expanded state.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

Special Instruction

The Nav Multi Vertical fragment offers a more complex version of this menu.

C.1.9 Nav Tabs Top

Adds a tabular style navigation bar that opens up to two levels of navigation. The tabs use individual images with rendered text (DHTML is not used in this fragment). You can choose a different theme, using different images, if you like.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

ssShowHome: Boolean flag that specifies whether to display a link to the home page. The default is "true."

ssShowNext: Boolean flag that specifies whether to display the second level of the navigation. The default is "false." If enabled, the second level of navigation opens below the tabs as text links, much like the Nav Plain Horizontal fragment.

ssTheme: Specifies the name of the theme. Specifically, it references a theme directory in the assets of this fragment.

C.1.10 Nav Wrap Horizontal

Adds a top-level navigation that can be split onto two lines, where the top line is given greater emphasis.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

ssShowHome: Boolean flag that specifies whether to display a link to the home page. The default is "false."

ssLineWrap: Specifies to break the display after the top-level heading. This is an integer value that defaults to 4. The Home link (if used) is not factored into this equation.

C.1.11 Search Box Plain

Adds a search box to the page template, using an HTML form to submit the search keywords. The search box must link to another page that executes the search and display the results.

Parameters

ssResultsPage: Specifies the page template that opens the search results (see the Search Result Plain fragment).

ssButtonText: The text for the "search" button, defaults to "Search."

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

Special Instruction

This fragment is intended for use with the Search Result Plain fragment or a similar, custom-built page that performs the search and display the results.

C.1.12 Search Result Paged

Adds a search results area to the page template. This fragment is designed for use with the Search Box Plain fragment. It takes a URL parameter "ssUserText" supplied by the search box. It combines that value with a restricting clause supplied as a parameter to this fragment. It executes the search and opens a batch of results as a paged list (instead of the **more...** button used in the Search Result Plain fragment) if needed.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssClassName: Specifies a CSS class name.

ssQueryText: This is used as a query restriction clause. It is combined with the ssUserText value received from the search box. If both components of the query are specified, the final query becomes: "(ssQueryText) AND (ssUserText)".

If combined also with ssLimitScope, the query becomes:

```
"(ssQueryText) AND (ssUserText) AND (ssLimitScope)"
```

If any component is blank/false, it is excluded from assembly.

ssLimitScope: Limits the scope of a search to within a web site.

ssSortField: The name of the metadata field to sort the results. The default is "dInDate".

ssSortOrder: The order to sort the results. Default is "desc".

ssResultCount: The number of results to show per page. The default is 25.

ssDisplayField: The name of the metadata field to be used to form the "label" of the link in the result.

ssTargetNodeID: Forces the target of a link so the contents display at a particular node, in the replaceable region of a secondary page. This is typically used for sharing content, where you include the same content item in multiple web sites.

Special Instruction

This fragment should be added to a section in your site hierarchy that is not included in your site navigation. It creates a placeholder page for the search results.

C.1.13 Search Result Plain

Adds a search results area to the page template. This fragment is designed for use with the Search Box Plain fragment. It takes a url parameter *ssUserText* supplied by the search box. It combines that value with a restricting clause supplied as a parameter to this fragment. It executes the search and opens a batch of results with a **more...** button if needed.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssClassName: Specifies a CSS class name.

ssQueryText: This is used as a query restriction clause. It is combined with the *ssUserText* value received from the search box. If both components of the query are specified, the final query becomes: "(ssQueryText) AND (ssUserText)".

If combined also with *ssLimitScope*, the query becomes:

```
"(ssQueryText) AND (ssUserText) AND (ssLimitScope) "
```

If any component is blank/false, it is excluded from assembly.

ssLimitScope: Limits the scope of a search to within a web site.

ssSortField: The name of the metadata field to sort the results. The default is "dInDate".

ssSortOrder: The order to sort the results. Default is "desc".

ssResultCount: The number of results to show per page. The default is 25.

ssDisplayField: The name of the metadata field to be used to form the "label" of the link in the result.

ssTargetNodeID: Forces the target of a link so the contents display at a particular node, in the replaceable region of a secondary page. This is typically used for sharing content, where you include the same content item in multiple web sites.

Special Instruction

This fragment should be added to a section in your site hierarchy that is not included in your site navigation. It creates a placeholder page for the search results.

C.1.14 Site Map Plain

Adds a site map that shows every section on the web site and opens it in a hierarchical fashion. The hierarchy is shown in its expanded state. This type of navigation can be especially useful as a navigation aid for visitors looking for information in a hurry.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

ssShowHome: Boolean flag that specifies whether to display a link to the home page. The default is "false."

ssShowFocus: Boolean flag that specifies whether to highlight the focus point, or current location in the site hierarchy. The default is "false."

C.1.15 Site Map Tree

Adds an expandable "tree view" of the complete site hierarchy. The tree view auto-expands to show the current location in the site hierarchy. You can then browse to other sections from there. (This fragment uses DHTML.)

Parameters

ssTheme: Specifies the name of the theme. Specifically, it references a theme directory in the assets of the fragment. The default is "explorer1."

ssShowHome: Boolean flag that specifies whether to display a link to the home page. The default is "true."

Special Instruction

The icons in the hierarchy use images stored in the assets of this fragment. You can choose a different theme, using different images, if you like.

C.1.16 Vertical Menu Bar

Adds a navigation menu that emulates a hierarchical menu bar that opens the complete site hierarchy. (This fragment uses DHTML.)

Parameters

ssShowHome: Boolean flag that specifies whether to display a link to the home page. The default is "true."

C.1.17 ASP Bread Crumb Plain

Adds navigational links that represent a hierarchical path, also called a breadcrumb. Breadcrumbs are frequently placed at the top or the bottom of a web page, and they help users orient themselves on a web site.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssClassName: Specifies a CSS class name.

ssSeparator: Text used to separate the links. The default is ">".

Special Instruction

This fragment is designed for ASP pages.

C.1.18 ASP Bread Crumb Plain (server)

Adds navigational links that represent a hierarchical path. This fragment is very similar to the ASP Bread Crumb Plain fragment: the difference being this one uses server-side script.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssClassName: Specifies a CSS class name.

ssSeparator: Text used to separate the links. The default is ">".

Special Instruction

This fragment is designed for ASP pages.

C.1.19 ASP Explorer Menu Bar

Adds a navigation menu that emulates a hierarchical menu bar, similar to that of a Windows-based application. (This fragment uses DHTML.)

Parameters

ssShowHome: Boolean flag that specifies whether to display a link to the home page. The default is "true."

Special Instruction

This fragment is designed for ASP pages.

C.1.20 ASP Nav Multi Horizontal

Adds a top navigation bar that can be placed anywhere on the page. This fragment can display a customized range of levels.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

ssSeparator: Text used to separate the parts of the bar. Default is "|".

ssShowHome: Boolean flag that specifies whether to display a link to the home page as part of the top level navigation. The default is "true." (The home page is actually located in the root of the web site, essentially above the "top" level, but it is perceived as a sibling.)

ssStartLevel: The display level to start displaying the menu.

ssNumLevels: The number of (section) levels to display.

Special Instruction

- The ASP Nav Plain Horizontal fragment offers a simpler version of this menu.
- This fragment is designed for ASP pages.

C.1.21 ASP Nav Multi Vertical

Adds a side navigation bar that can go anywhere on the page template, but is generally placed on the left side where it indents from left to right. This fragment can display a customized range of levels (unlike the Nav Plain Vertical fragment, which

opens only the second-level). It opens at the current focus level, so if your site does not have the sub-level sections that you specify in this fragment, then nothing opens.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

ssShowHome: Boolean flag that specifies whether to display a link to the home page as part of the top-level navigation. The default is "false."

ssStartLevel: The display level to start displaying the menu.

ssNumLevels: The number of (section) levels to display.

Special Instruction

This fragment is designed for ASP pages.

C.1.22 ASP Search Box Plain

Adds a search box to the page template, using an HTML form to submit the search keywords. The search box must link to another page that executes the search and display the results.

Parameters

ssResultsPage: Specifies the page template that opens the search results (see the ASP Search Result Plain fragment).

ssButtonText: The text for the "search" button, defaults to "Search."

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

Special Instruction

- This fragment is intended for use with the ASP Search Result Plain fragment or a similar, custom-built page that performs the search and display the results.
- This fragment is designed for ASP pages.

C.1.23 ASP Search Box Plain (server)

Adds a search box to the page template. The search box must link to another page that executes the search and display the results. This fragment is very similar to the ASP Search Box Plain fragment: the difference being this one uses server-side script.

Parameters

ssResultsPage: Specifies the page template that opens the search results (see the Search Result Plain fragment).

ssButtonText: The text for the "search" button, defaults to "Search."

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

Special Instruction

- This fragment is intended for use with the ASP Search Result Plain fragment or a similar, custom-built page that performs the search and display the results.
- This fragment is designed for ASP pages.

C.1.24 ASP Search Result Plain

Adds a search results area to the page template. This fragment is designed for use with the ASP Search Box Plain fragment. It takes a url parameter *ssUserText* supplied by the search box. It combines that value with a restricting clause supplied as a parameter to this fragment. It executes the search and opens a batch of results with a **more...** button if needed.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssClassName: Specifies a CSS class name.

ssQueryText: This is used as a query restriction clause. It is combined with the *ssUserText* value received from the search box. If both components of the query are specified, the final query becomes: "(*ssQueryText*) AND (*ssUserText*)".

If combined also with *ssLimitScope*, the query becomes:

```
"(ssQueryText) AND (ssUserText) AND (ssLimitScope)"
```

If any component is blank/false, it is excluded from assembly.

ssLimitScope: Limits the scope of a search to within a web site.

ssSortField: The name of the metadata field to sort the results. The default is "dInDate".

ssSortOrder: The order to sort the results. Default is "desc".

ssResultCount: The number of results to show per page. The default is 25.

ssDisplayField: The name of the metadata field to be used to form the "label" of the link in the result.

ssTargetNodeID: Forces the target of a link so the contents display at a particular node, in the replaceable region of a secondary page. This is typically used for sharing content, where you include the same content item in multiple web sites.

Special Instruction

- This fragment should be added to a section in your site hierarchy that is not included in your site navigation. It creates a placeholder page for the search results.
- This fragment is designed for ASP pages.

C.1.25 ASP Site Map Plain

Adds a site map that shows every section on the web site and opens it in a hierarchical fashion. The hierarchy is shown in its expanded state. This type of navigation can be especially useful as a navigation aid for visitors looking for information in a hurry.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

ssShowHome: Boolean flag that specifies whether to display a link to the home page. The default is "false."

ssShowFocus: Boolean flag that specifies whether to highlight the focus point, or current location in the site hierarchy. The default is "false."

Special Instruction

This fragment is designed for ASP pages.

C.1.26 CSP Sample Navigation (client)

A sample navigation that shows how to access custom section properties using client-side script. This sample provides only top-level links for the hierarchy, and does not link to child sections. Each link can be customized.

This adds customizable navigation options so you can tailor the site navigation display. For example, you can display different colors for each section, replace the text that displays, or replace text with an image.

Special Instruction

This assumes the existence of the following four custom section properties:

Name: cspNavInclude

Type: boolean

Description: This shows an example of how to override the default "Include Section in Navigation" behavior. True or false value, to decide if this section should be shown in navigation.

Name: cspNavLabel

Type: Text

Description: An optional label to override the default behavior of displaying the section name in the navigation presentation.

Name: cspNavImage

Type: managedurl

Description: An optional url of an image file to be used in the navigation, instead of the text label.

Name: cspNavColor

Type: Color

Description: An optional override to specify the lead color of the left-hand table cell in the navigation fragment for each section.

Note: When using CSP (custom section property) fragments, you must define the properties, and then set the values for each custom section property in every section that contains the fragment.

C.1.27 CSP Sample Navigation (server)

A sample navigation that determines how to access custom section properties using server-side script. This sample provides only top-level links for the hierarchy, and does not link to child sections. Each link can be customized.

This adds customizable navigation options so you can tailor the site navigation display. For example, you can display different colors for each section, replace the text that displays, or replace text with an image.

Special Instruction

Same as CSP Sample Navigation (client). For more information, see [Section C.1.26, "CSP Sample Navigation \(client\)."](#)

Note: When using CSP (custom section property) fragments, you must define the properties, and then set the values for each custom section property in every section that contains the fragment.

C.1.28 JSP Nav Home Logo

Adds an image that serves as a hyperlink to the home page of the web site.

Parameters

ssLogo: Specifies the URL of the graphic to use in the content server. If this setting is left blank, a placeholder image called "logo.gif" from the fragment assets directory is used.

ssWidth: The width of the image (in pixels). If left blank, no width attribute is used.

ssHeight: The height of the image (in pixels). If left blank, no height attribute is used.

Special Instruction

This fragment is designed for JSP page templates.

C.1.29 JSP Nav Plain Horizontal

Adds a top navigation bar that can be placed anywhere on the page. This fragment opens only the top level of your site navigation; that is, the sections in the root of the site.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

ssSeparator: Text used to separate the parts of the bar. Default is "|".

ssShowHome: Boolean flag that specifies whether to display a link to the home page. The default is "true." (The home page is actually located in the root of the web site, above the "top" level, but it may be perceived as a sibling here.)

Special Instruction

This fragment is designed for JSP page templates.

C.1.30 JSP Nav Plain Vertical

Adds a side navigation bar that can go anywhere on the page template, but is generally placed on the left side where it indents from left to right. This fragment opens only the second-level of the site navigation, so if your site has no second-level sections then nothing opens. The site hierarchy opens in its expanded state.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

Special Instruction

This fragment is designed for JSP page templates.

C.1.31 JSP Search Box Plain

Adds a search box to the page template. The search box must link to another page that executes the search and opens the results.

Parameters

ssResultsPage: Specifies the page template that opens the search results (see the Search Result Plain fragment).

ssButtonText: The text for the "search" button, defaults to "Search."

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

Special Instruction

This fragment is intended for use with the JSP Search Result Plain fragment or a similar, custom-built page that performs the search and opens the results.

This fragment is designed for JSP page templates.

C.1.32 JSP Search Results Paged

Adds a search results area to the page template. This fragment is designed for use with the Search Box Plain fragment. It takes a url parameter *ssUserText* supplied by the search box. It combines that value with a restricting clause supplied as a parameter to this fragment. It executes the search and opens a batch of results as a paged list (instead of the **more...** button used in the Search Result Plain fragment) if needed.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssClassName: Specifies a CSS class name.

ssQueryText: This is used as a query restriction clause. It is combined with the `ssUserText` value received from the search box. If both components of the query are specified, the final query becomes:

```
"(ssQueryText) AND (ssUserText) "
```

If combined also with `ssLimitScope`, the query becomes:

```
"(ssQueryText) AND (ssUserText) AND (ssLimitScope) "
```

If any component is blank/false, it is excluded from assembly.

ssLimitScope: Limits the scope of a search to within a web site.

ssSortField: The name of the metadata field to sort the results. The default is "dInDate".

ssSortOrder: The order to sort the results. Default is "desc".

ssResultCount: The number of results to show per page. The default is 25.

ssDisplayField: The name of the metadata field to be used to form the "label" of the link in the result.

ssTargetNodeID: Forces the target of a link so the contents display at a particular node, in the replaceable region of a secondary page. This is typically used for sharing content, where you include the same content item in multiple web sites.

Special Instruction

- This fragment should be added to a section in your site hierarchy that is not included in your site navigation. It creates a placeholder page for the search results.
- This fragment is designed for JSP page templates.

C.1.33 JSP Site Map Plain

Adds a site map that shows every section on the web site and opens it in a hierarchical fashion. The hierarchy is shown in its expanded state. This type of navigation can be especially useful as a navigation aid for visitors looking for information in a hurry.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

ssShowHome: Boolean flag that specifies whether to display a link to the home page. The default is "false."

ssShowFocus: Boolean flag that specifies whether to highlight the focus point, or current location in the site hierarchy. The default is "false."

Special Instruction

This fragment is designed for JSP page templates.

C.1.34 IDOC Breadcrumb Plain

Adds navigational links that represent a hierarchical path, also called a breadcrumb. Breadcrumbs are frequently placed at the top or the bottom of a web page, and they help users orient themselves on a web site. This fragment is very similar to the Breadcrumb Plain fragment: the difference being this one uses Idoc Script and relies heavily on service calls in the content server.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssClassName: Specifies a CSS class name.

ssSeparator: Text used to separate the links. The default is ">".

C.1.35 IDOC Nav Home Logo

Adds an image that serves as a hyperlink to the home page of the web site. This fragment is very similar to the Nav Home Logo fragment: the difference being this one uses Idoc Script and relies heavily on service calls in the content server.

Parameters

ssLogo: Specifies the URL of the graphic to use in the content server. If this setting is left blank, a placeholder image called "logo.gif" from the fragment assets directory is used.

ssWidth: The width of the image (in pixels). If left blank, no width attribute is used.

ssHeight: The height of the image (in pixels). If left blank, no height attribute is used.

C.1.36 IDOC Search Box Plain

Adds a search box to the page template, using an HTML form to submit the search keywords. The search box must link to another page that executes the search and opens the results. This fragment is very similar to the Search Box Plain fragment: the difference being this one uses Idoc Script and relies heavily on service calls in the content server.

Parameters

ssResultsPage: Specifies the page template that opens the search results (see the Search Result Plain fragment).

ssButtonText: The text for the "search" button, defaults to "Search."

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

Special Instruction

This fragment is intended for use with the IDOC Search Result Plain fragment or a similar, custom-built page that performs the search and opens the results.

C.1.37 RSET Breadcrumb Plain

Adds navigational links that represent a hierarchical path. This fragment is very similar to the Breadcrumb Plain fragment: the difference being this one uses Idoc Script and uses values from a result set. Compare IDOC Breadcrumb Plain.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssClassName: Specifies a CSS class name.

ssSeparator: Text used to separate the links. The default is ">".

C.1.38 RSET Nav Multi Horizontal

Adds a top navigation bar that can be placed anywhere on the page. This fragment is very similar to the Nav Multi Horizontal fragment: the difference being this one uses Idoc Script and uses values from a result set. Compare IDOC Nav Multi Horizontal.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

ssSeparator: Text used to separate the parts of the bar. Default is "|".

ssShowHome: Boolean flag that specifies whether to display a link to the home page as part of the top level navigation. The default is "true." (The home page is actually located in the root of the web site, essentially above the "top" level, but it is perceived as a sibling.)

ssStartLevel: The display level to start displaying the menu.

ssNumLevels: The number of (section) levels to display.

Special Instruction

The RSET Nav Plain Horizontal fragment offers a simpler version of this menu.

C.1.39 RSET Nav Plain Horizontal

Adds a top navigation bar that can be placed anywhere on the page. This fragment is very similar to the Nav Plain Horizontal fragment: the difference being this one uses Idoc Script and uses values from a result set. Compare IDOC Nav Plain Horizontal.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

ssSeparator: Text used to separate the parts of the bar. Default is "|".

ssShowHome: Boolean flag that specifies whether to display a link to the home page. The default is "true." (The home page is actually located in the root of the web site, above the "top" level, but it may be perceived as a sibling here.)

Special Instruction

The RSET Nav Multi Horizontal fragment offers a more complex version of this menu.

C.1.40 RSET Site Map Plain

Adds a site map that shows every section on the web site and opens it in a hierarchical fashion. The hierarchy is shown in its expanded state. This fragment is very similar to the Site Map Plain fragment: the difference being this one uses Idoc Script and uses values from a result set. Compare IDOC Site Map Plain.

Parameters

ssClassName: Specifies a CSS class name.

C.1.41 Additional IDOC Fragments

There are some IDOC fragments from Oracle Site Studio releases before 10gR4 that have some functions that could be useful. However, it is important to note that these fragments do not scale for large applications, and as such are not recommended for use.

These IDOC fragments are installed with Oracle Site Studio, but are not checked in:

- [Section C.1.41.1, "IDOC Nav Multi Horizontal"](#)
- [Section C.1.41.2, "IDOC Nav Plain Horizontal"](#)
- [Section C.1.41.3, "IDOC Nav Tabs Top"](#)
- [Section C.1.41.4, "IDOC Site Map Plain"](#)

C.1.41.1 IDOC Nav Multi Horizontal

Adds a top navigation bar that can be placed anywhere on the page. This fragment can display a customized range of levels. This fragment is very similar to the Nav Multi Horizontal fragment: the difference being this one uses Idoc Script and relies heavily on service calls in the content server.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

ssSeparator: Text used to separate the parts of the bar. Default is "|".

ssShowHome: Boolean flag that specifies whether to display a link to the home page as part of the top level navigation. The default is "true." (The home page is actually located in the root of the web site, essentially above the "top" level, but it is perceived as a sibling.)

ssStartLevel: The display level to start displaying the menu.

ssNumLevels: The number of (section) levels to display.

Special Instruction

The IDOC Nav Plain Horizontal fragment offers a simpler version of this menu.

C.1.41.2 IDOC Nav Plain Horizontal

Adds a top navigation bar that can be placed anywhere on the page. This fragment opens only the top level of your site navigation; that is, the sections in the root of the

site. This fragment is very similar to the Nav Plain Horizontal fragment: the difference being this one uses Idoc Script and relies heavily on service calls in the content server.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

ssSeparator: Text used to separate the parts of the bar. Default is "|".

ssShowHome: Boolean flag that specifies whether to display a link to the home page. The default is "true." (The home page is actually located in the root of the web site, above the "top" level, but it may be perceived as a sibling here.)

Special Instruction

The IDOC Nav Multi Horizontal fragment offers a more complex version of this menu.

C.1.41.3 IDOC Nav Tabs Top

Adds a tabular style navigation bar that opens up to two levels of navigation. The tabs use individual images with rendered text (DHTML is not used in this fragment). You can choose a different theme, using different images, if you like. This fragment is very similar to the Nav Tabs Top fragment: the difference being this one uses Idoc Script and relies heavily on service calls in the content server.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssFocusColor: Changes the color of the text when the link is selected.

ssClassName: Specifies a CSS class name.

ssShowHome: Boolean flag that specifies whether to display a link to the home page. The default is "true."

ssShowNext: Boolean flag that specifies whether to display the second level of the navigation. The default is "false." If enabled, the second level of navigation opens below the tabs as text links, much like the Nav Plain Horizontal fragment.

ssTheme: Specifies the name of the theme. Specifically, it references a theme directory in the assets of this fragment.

C.1.41.4 IDOC Site Map Plain

Adds a site map that shows every section on the web site and opens it in a hierarchical fashion. The hierarchy is shown in its expanded state. This fragment is very similar to the Site Map Plain fragment: the difference being this one uses Idoc Script and relies heavily on service calls in the content server.

Parameters

ssClassName: Specifies a CSS class name.

C.2 Dynamic List Fragments

These are the dynamic list fragments:

- [Section C.2.1, "Dynamic List Generic"](#)
- [Section C.2.2, "Dynamic List Paged"](#)
- [Section C.2.3, "Dynamic List Plain"](#)
- [Section C.2.4, "ASP Dynamic List Paged"](#)
- [Section C.2.5, "ASP Dynamic List Simple"](#)
- [Section C.2.6, "CSP Sample Dynamic List"](#)
- [Section C.2.7, "JSP Dynamic List Plain"](#)

C.2.1 Dynamic List Generic

Adds a dynamic list in a contribution region on the page template. A dynamic list performs a search in the content server and opens everything matching particular criteria. Contributors can add to the list by checking in new content items with metadata matching the search criteria.

Parameters

Many of the parameters in this fragment are the same as those found in the Dynamic List Plain fragment (see [Section C.2.3, "Dynamic List Plain"](#)).

C.2.2 Dynamic List Paged

Similar to Dynamic List Plain, but offers a "results page list" like:

[[prev](#)] [1](#) | [2](#) | [3](#) | [4](#) | [5](#) [[next](#)]

The above example shows what the user would see on the second page of the results.

Parameters

The same as Dynamic List Plain, except for the following differences:

ssWantMore: Boolean flag that specifies whether to include a "results page list" when necessary. The default is "true." (Set this false if you want to limit one batch of results.)

ssMoreText: Not used.

ssTargetNodeID: Forces the target of a link so the contents display at a particular node, in the replaceable region of a secondary page. This is typically used for sharing content, where you include the same content item in multiple web sites.

C.2.3 Dynamic List Plain

Adds a dynamic list in a contribution region on the page template. A dynamic list performs a search in the content server and opens everything matching particular criteria. Contributors can add to the list by checking in new content items with metadata matching the search criteria.

This dynamic list sample divides the search results page into several sections, presenting each of them in a separated <div> tag and allowing parameters to specify plain text (or elaborate HTML) to display in each section.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssClassName: Specifies a CSS class name.

ssQueryText: Specifies the query that is performed in the content server.

ssLimitScope: Limits the scope of a search to within a web site.

ssSortField: The name of the metadata field to sort the results. The default is "dInDate".

ssSortOrder: The order to sort the results. The default is "desc".

ssResultCount: The number of results to show per page. The default is 25.

ssCaptionText: Caption text to appear in the head of the results display. The default is "List Results".

ssReportText: Report text that opens below the caption and summarizes the results. The default is the following, where "TotalRows" is an Idoc variable standard for executing a search, and the "ssFirstHit" and "ssLastHit" variables are calculated in the fragment code to make the display easier:

```
Total: <${TotalRows}>
Items: <${ssFirstHit}> - <${ssLastHit}>
```

ssDisplayStart: Text to display at the start of the core list. The default is "".

ssRowStart: Text to display at the start of each row. The default is "".

ssRowLink: Text to display as the label of the "link" in each row. The default is "<\${dDocTitle}>".

ssRowEnd: Text to display at the end of each row. The default is "".

ssDisplayEnd: Text to display at the end of the core list. The default is "".

ssWantMore: Boolean flag that specifies whether to include a "more..." button when necessary. The default is "true." (To limit the search to one batch of results, set this to "false.")

ssMoreText: Text to display as the label of the "more..." link when necessary. The default is "more..."

ssNoResults: Text to display when the search produces no results. The default is "No items match your search criteria".

ssTargetNodeID: Forces the target of a link so the contents display at a particular node, in the replaceable region of a secondary page. This is typically used for sharing content, where you include the same content item in multiple web sites.

C.2.4 ASP Dynamic List Paged

Similar to Dynamic List Plain, but offers a "results page list" like:

```
[prev] 1 | 2 | 3 | 4 | 5 [next]
```

The above example shows what the user would see on the second page of the results.

Parameters

The same as for the Dynamic List Plain fragment (see [Section C.2.3, "Dynamic List Plain"](#)), except for the following differences:

ssWantMore: Boolean flag that specifies whether to include a "results page list" when necessary. The default is "true." (Set this false if you want to limit one batch of results.)

C.2.5 ASP Dynamic List Simple

Similar to Dynamic List Simple, but used for ASP.

Parameters

ssTargetNodeID: Forces the target of a link so the contents display at a particular node, in the replaceable region of a secondary page. This is typically used for sharing content, where you include the same content item in multiple web sites.

C.2.6 CSP Sample Dynamic List

A sample dynamic list fragment used with custom section properties. Adds a dynamic list that gives you the ability to use a different query in each section, instead of using one query in all of the sections that use the page.

Parameters

ssQueryText: Specifies the query that is performed in the content server.

ssSortField: The name of the metadata field to sort the results. The default is "dInDate".

ssSortOrder: The order to sort the results. The default is "desc".

ssResultCount: The number of results to show per page. The default is 25.

ssLimitScope: Limits the scope of a search to within a web site.

Special Instruction

This sample assumes the existence of the following custom section property:

Name: cspQueryText

Type: managedquery

Description: An optional query text string which is appended to the regular query text parameter, if any, so each section can define a unique query text.

Note: When using CSP (custom section property) fragments, you must define the properties, and then set the values for each custom section property in every section that contains the fragment.

C.2.7 JSP Dynamic List Plain

Adds a dynamic list in a contribution region on the page template. A dynamic list performs a search in the content server and opens everything matching particular criteria. Contributors can add to the list by checking in new content items with metadata matching the search criteria.

This dynamic list sample divides the search results page into several sections, presenting each of them in a separated <div> tag and allowing parameters to specify plain text (or elaborate HTML) to display in each section.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssClassName: Specifies a CSS class name.

ssQueryText: Specifies the query that is performed in the content server.

ssLimitScope: Limits the scope of a search to within a web site.

ssSortField: The name of the metadata field to sort the results. The default is "dInDate".

ssSortOrder: The order to sort the results. The default is "desc".

ssResultCount: The number of results to show per page. The default is 25.

ssDisplayField: The name of the metadata field to be used to form the "label" of the link in the result.

ssTargetNodeID: Forces the target of a link so the contents display at a particular node, in the replaceable region of a secondary page. This is typically used for sharing content, where you include the same content item in multiple web sites.

Special Instruction

This fragment is designed for JSP page templates.

C.3 Static List Fragments

These are the static list fragments:

- [Section C.3.1, "Static List Example"](#)
- [Section C.3.2, "Static List Generic"](#)
- [Section C.3.3, "ASP Static List Example"](#)

C.3.1 Static List Example

Adds a static list with four columns in a contribution region on the page template. A static list comprises a fixed number of columns, each containing text or graphics, which can be edited in Contributor.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssClassName: Specifies a CSS class name.

C.3.2 Static List Generic

Adds a static list in a contribution region on the page template. A static list comprises a fixed number of columns, each containing text or graphics, which can be edited in Contributor.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssClassName: Specifies a CSS class name.

Column1_Name (1-6): Specifies the name of the column. This should be concise and should not contain spaces or special characters.

Column1_Label (1-6): Specifies the label of the column. This should be concise and should not contain spaces or special characters.

ssCaptionText: Opens the caption for the page.

ssDisplayStart: Specifies the start of the display loop.

ssDisplayRow: Opens a single row.

ssDisplayEnd: Specifies the end of the display loop.

ssBottomText: Specifies the button on the page.

C.3.3 ASP Static List Example

Adds a static list with four columns in a contribution region on the page template. A static list comprises a fixed number of columns, each containing text or graphics, which can be edited in Contributor.

Parameters

ssTextColor: Changes the color of the text.

ssHoverColor: Changes the color of the mouseover text.

ssClassName: Specifies a CSS class name.

C.4 Other Fragments

These fragments provide various additional functionality or examples:

- [Section C.4.1, "Copyright Example"](#)
- [Section C.4.2, "Dynamic Conversion"](#)
- [Section C.4.3, "Dynamic Conversion IFrame"](#)
- [Section C.4.4, "Error Handler Message"](#)
- [Section C.4.5, "Flash Example"](#)
- [Section C.4.6, "Logo Example"](#)
- [Section C.4.7, "Random Conversion"](#)
- [Section C.4.8, "ASP Error Handler Message"](#)
- [Section C.4.9, "CSP Sample Page Title"](#)
- [Section C.4.10, "PageHitCount"](#)
- [Section C.4.11, "Oracle Site Studio Manager"](#)

C.4.1 Copyright Example

Inserts a copyright into the page template, similar to the Logo Example fragment.

Parameters

No parameters.

C.4.2 Dynamic Conversion

Adds a dynamically converted native document to your page template using Dynamic Converter. When you select a native document, you can specify a GUI Template or a Layout Template for the conversion. If you do not specify either, a special conversion is used with an empty conversion template and layout template.

Parameters

ssDocName: The dDocName of the document to be converted.

ssTemplate: The dDocName of the Dynamic Converter GUI Template.

ssLayout: The dDocName of the Dynamic Converter Layout Template.

Special Instruction

You can also add native documents to your site by assigning one to a region, creating links to them, and allowing contributors to add and edit them.

C.4.3 Dynamic Conversion IFrame

Similar to the Dynamic Conversion fragment (see [Section C.4.2, "Dynamic Conversion"](#)), but adds an IFrame allowing you to scroll through the document.

Parameters

ssDocument: The dDocName of the document to be converted.

ssTemplate: The dDocName of the Dynamic Converter GUI Template.

ssLayout: The dDocName of the Dynamic Converter Layout Template.

ssHeight: The height of the IFrame in pixels or percent.

ssWidth: The width of the IFrame in pixels or percent.

ssBorder: Select '1' for border or '0' for no border.

ssScrolling: Scroll bars can be set to 'yes', 'no', or 'auto' (when required).

Special Instruction

You can also add native documents to your site by assigning one to a region, creating links to them, and allowing contributors to add and edit them.

C.4.4 Error Handler Message

Adds the ability to display content server error messages (Idoc message variables) in a section of your web site.

Parameters

No parameters.

Special Instruction

This fragment can be used when you select a web site section to display content server errors (see [Section 12.1.9, "Designating a Section as an Error Handler"](#)).

C.4.5 Flash Example

Adds a Flash presentation to your page template.

Parameters

ssAnimation: Use to locate the Flash presentation in the content server.

sswidth: Use to specify the width of the Flash animation.

ssheight: Use to specify the height of the Flash animation.

C.4.6 Logo Example

Adds a logo to your page template as a referenced object to illustrate how you can manage content separate from the page template.

Parameters

No parameters.

C.4.7 Random Conversion

Adds a randomly converted native document to your page template; especially useful for web banners. The conversion requires Dynamic Converter. When you select a native document, you can specify a GUI Template or a Layout Template for the conversion. If you do not specify either, a special conversion takes place using a blank conversion template and an empty layout template.

Parameters

ssQueryText: Specifies the complete query clause.

ssLimitScope: Limits the scope of a search to within a web site.

ssTemplate: The dDocName of the Dynamic Converter GUI Template.

ssLayout: The dDocName of the Dynamic Converter Layout Template.

C.4.8 ASP Error Handler Message

Adds the ability to display content server error messages (message variables) in a section of your web site.

Parameters

No parameters.

Special Instruction

This fragment can be used when you select a web site section to display web site errors (see [Section 12.1.9, "Designating a Section as an Error Handler"](#)).

C.4.9 CSP Sample Page Title

Adds the ability to change the <title> tag that each section opens on a web site, for each section that uses that page template.

Special Instruction

This assumes the existence of the following custom section property:

Name: cspPageTitle

Type: Text

Description: Use this as the page title when viewing this section instead of the value of the <title> tag in the layout.

Note: When using CSP (custom section property) fragments, you must define the properties, and then set the values for each custom section property in every section that contains the fragment.

C.4.10 PageHitCount

Fragment that demonstrates integration between Content Tracker and Oracle Site Studio.

Parameters

contributorOnly: Boolean flag that specifies whether the hit count should be shown all the time or only in contribution mode. The default is "false" (show hit count all the time).

Note: This feature works best on HCSP/Idoc web sites using SQL Server.

C.4.11 Oracle Site Studio Manager

Allows users to manage the site hierarchy using a web-based application called Oracle Site Studio Manager. For more information, see [Chapter 16, "Setting Up Manager,"](#) and *Managing Oracle Site Studio*.

Parameters

theme: Look and feel themed CSS. The default is "blue."

settings: Oracle Site Studio Manager configuration settings file.

Design Mode

Design mode is a subset of some of the functions available in Oracle Site Studio Designer that are made accessible through Oracle Site Studio Contributor. While in Oracle Site Studio Designer, you can set which role or roles can access design mode, and those roles can remotely make changes to the placeholders and site asset relationships.

This section covers the following topics:

- [Section D.1, "About Design Mode"](#)
- [Section D.2, "Making Design Mode Available"](#)
- [Section D.3, "Using Design Mode"](#)
- [Section D.4, "Using the Placeholder Section Properties Dialog"](#)

D.1 About Design Mode

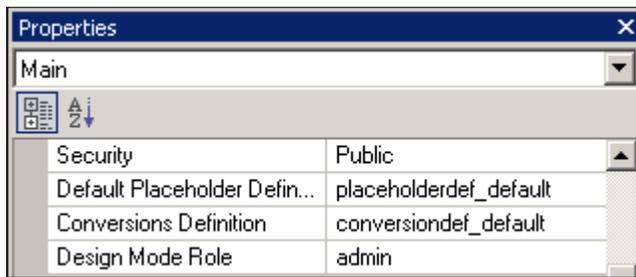
Design mode is a subset of some of the functions available in Oracle Site Studio Designer that are made accessible through Oracle Site Studio Contributor. While in Designer, you can set which role or roles can access design mode, and those roles can remotely make changes to the placeholders and site asset relationships.

Design mode does not allow changes to the site assets themselves; you cannot change a data file the way you can in contribution mode. However, you can change the data file, definitions, or templates used, as well as work with the replaceable region on secondary pages.

D.2 Making Design Mode Available

The user roles that have access to design mode are set in the properties pane for the web site or section of a web site in Designer.

It is important to note that the role is set, not a specific user.

Figure D–1 The Design Mode Role in the Properties Pane

When the design mode role is set, it is applied site-wide. You cannot limit design mode access to certain sections.

D.3 Using Design Mode

Design mode is useful for those who need to work with Designer but do not have access to the Oracle Site Studio Designer application. A typical instance of this is when a user is using the Oracle Site Studio extension for JDeveloper.

It is important to remember that when you have entered design mode, that you cannot edit an individual site asset. You can only change the relationship of the assets, such as using a different region template, or placeholder definition, or defining the placeholder section properties for the section including related site assets. If you want to edit the content in a data file, you must be in contribution mode. If you want to change the properties of the templates or definitions themselves, then you must do so in Oracle Site Studio Designer.

This section covers the following topics:

- [Section D.3.1, "Entering Design Mode"](#)
- [Section D.3.2, "Working with Primary Pages"](#)
- [Section D.3.3, "Working with Secondary Pages"](#)

D.3.1 Entering Design Mode

You enter design mode by clicking the **Design Mode** icon ([Figure D–2](#)) after entering contribution mode.

Design mode is entered after you have already entered contribution mode.

Figure D–2 Design Mode Icon

The **Design Mode** icon is located in the contribution mode toolbar. It will appear only if your role has been specified in Designer.

When you enter design mode, the toolbar will change to show three icons ([Figure D–3](#)).

Figure D–3 Primary Page Design Mode Toolbar

The left-most icon is the **Contribution Mode** icon, and it returns you to contribution mode.

The middle icon is the **Show Placeholder Section Properties** icon. This is used to open the Placeholder Section Properties dialog, where you can add, edit, or delete placeholders on the page. For more information on the Placeholder Section Properties dialog, see [Section D.4, "Using the Placeholder Section Properties Dialog."](#)

The last icon takes you to the secondary page for the section. If the section does not have a secondary page, the icon will not be present. When you go to the secondary page, the toolbar will change slightly ([Figure D-4](#)).

Figure D-4 Secondary Page Design Mode Toolbar



The toolbar for the secondary page has the same contribution mode and Placeholder Section Properties dialog.

You can use the difference in icons to help you keep track of whether you are on a primary or secondary page. In the toolbar on the primary page, there is a "2" in the right-most icon. This signifies taking you to the secondary page. In the secondary page toolbar, there is a "1", which signifies taking you to the primary page.

D.3.2 Working with Primary Pages

When you first enter design mode from contribution mode, the bar at the top of the page will change from Contribution Mode to Design Mode, and the Primary Page toolbar will appear.

The menu for the contribution graphics will be slightly different. There will be up to four options, depending on the settings for reports:

- Switch Content
- Switch Region Template
- Apply to All Sections
- View Web Site Usage Report

The **Switch Content**, **Switch Region Template**, and **View Web Site Usage Report** (when made available) menu options are the same as they would be in contribution mode.

The **Apply to All Sections** menu option is used to apply the settings of the current section's placeholder section properties to all other sections that use the page template.

D.3.3 Working with Secondary Pages

Design mode allows you to view the secondary page, which is not possible in contribution mode without using explicit links.

The replaceable regions on the secondary page will have a contribution graphic showing that they are replaceable ([Figure D-5](#)).

Figure D-5 Contribution Graphic Showing Replaceable Region



- Switch Content
- Switch Region Template
- Set Replaceable
- Clear Replaceable
- View Web Site Usage Report

The **Switch Content**, **Switch Region Template**, and **View Web Site Usage Report** (when made available) menu options are the same as they would be in contribution mode.

The **Set Replaceable** menu option is used to set the selected placeholder as a replaceable region. Similarly, the **Clear Replaceable** menu option is used to clear the replaceable region settings from the placeholder.

D.4 Using the Placeholder Section Properties Dialog

The Placeholder Section Properties dialog is used primarily to edit the name of a placeholder, or change the region template, subtemplate, data file, or placeholder definition used in a placeholder.

Since this dialog only changes the placeholder section properties, a new placeholder can be named and have a region template, placeholder definition, and data file associated, but this information will only be stored in the placeholder section properties. It will not actually change the page template to create a new placeholder.

The dialog can also be used to delete placeholder section properties, if desired.

To modify the placeholder section properties, follow these steps:

1. On either a primary or secondary page, click the **Show Placeholder Section Properties** icon. It is located in the center of the toolbar.

The Placeholder Section Properties dialog opens.

2. If you want to remove a placeholder section property, select the row of the placeholder you want to delete, and click the **Delete Row** icon. (Figure D-6)

Figure D-6 Delete Icon



If you want to add a placeholder section property, click the **Add Row** icon. (Figure D-7)

Figure D-7 Add Row Icon



If you want to modify a placeholder section property, you can select the property and click the **Edit Row** icon (Figure D-8), or you can double-click the property.

Figure D-8 Edit Row Icon



When you edit, the Placeholder Section Mappings dialog opens.

3. In the Placeholder Section Mappings dialog, you can edit any of the values. Press **Select** to open a window displaying the results of a content server search, or simply enter the value you want.
4. Click **OK** to save the values on the Placeholder Section Mappings dialog and close the dialog.
5. Click **OK** to save the values on the Placeholder Section Properties dialog and close the dialog.

Glossary

The following terms are frequently used in the Oracle Site Studio suite of applications. They are listed here in alphabetical order.

address

See **site address**.

administrator

An individual who uses the performs the administrative Oracle Site Studio tasks on the content server. This would include tasks such as assigning web addresses to the site, backing up the site, replicating the site, and so on.

See also: **designer, contributor, manager**.

Cascading Style Sheet (CSS)

A file that provides control over how page content is displayed. More specifically, how different HTML elements, such as headers and links, appear on the page. A means of separating structure from presentation to control the formatting and layout of content in one place. Style sheets can be included in an HTML document by linking to an outside style sheet, embedding a document-wide style in the <HEAD> tag of the document, or embedding inline styles where needed.

content server

The central repository of all files associated with an Oracle Site Studio web site. The server provides advanced content management features such as library services (check-in, check-out, and the like), versioning, workflow, content conversion, and more.

contribution graphic

The graphic that displays beside each contribution region on the web page, when the page is viewed in contribution mode. Clicking the edit icon opens Contributor (or a third-party application for a native document), enabling contributors to edit the region content assigned to the region. Clicking the menu icon provides more options, such as switching the file assigned to the region and approving or rejecting a document for workflow.

contribution mode

A way of viewing a web page in a web browser whereby you can see and edit the contribution regions (and perhaps instances of Manager) on the page.

When the page is viewed in contribution mode, you see a contribution graphic beside each editable region. Click the graphic to edit the contents of the region (a data file or native document), approve it or reject the contents (when workflow is used), update

its metadata, and so on. You enter contribution mode using a combination of keystrokes (the default is Ctrl+Shift+F5).

contribution region

An area where contributors can add and edit content on a web page, as defined by the site designer. When the web page is viewed in contribution mode, contributors can edit the content in the region using either Contributor (for contributor data files) or the application associated with the content (for native documents).

See also: **placeholder**, **replaceable region**.

Contributor

One of three applications in Oracle Site Studio. Contributor is used to edit contributor data files associated with a contribution region. The way you set up a contribution region and contribution elements in Designer affects the appearance and behavior of the Contributor application. Typically, you have many different contributors using the Contributor application.

See also: **Designer**, **Manager**.

contributor

An individual who edits the content on an Oracle Site Studio web site, using either Contributor or a third-party application. The contributor can be anyone familiar with the content on the web site (a manager, a member of the sales or marketing team, a member of human resources, and so on). Using Oracle Site Studio, you often have multiple contributors working with a handful of managers and one designer.

See also: **designer**, **administrator**, **manager**.

contributor data file

An XML file created by Oracle Site Studio and designed to be opened and edited using the Contributor application. A contributor data file stores site content, and can be assigned to a contribution region, where it appears on the web site. The file itself, however, is stored as a managed content item in the content server, separate from the page templates, region templates, subtemplates, and other managed site assets. This enables content (created by contributors) to be separate from presentation (created by site designers).

See also: **native document**.

contributor-only section

A section in the site hierarchy that only appears on the site when it is viewed in contribution mode. Contributor-only sections can be useful to convey information, like instructions for contributing to the site, from designers to managers, managers to contributors, and so on.

conversion definition

A file that specifies the conversion rules for native documents on a web site. Each conversion definition is a rule, template, or combination that defines how a native document is converted to HTML and displayed in the page.

conversion template

A part of Dynamic Converter used in Oracle Site Studio to define how native documents are converted to HTML for viewing in a web browser. Native documents on the content server are converted based on their metadata and file type.

CSS

See **Cascading Style Sheet (CSS)**.

custom element

A type of element that uses a custom element form. A custom element is quite different from the other elements (WYSIWYG, plain text, dynamic list, static list, and image) in that it is used to create a custom interface (web-based form) in the Contributor application. Contributors use the interface to add various types of content and perform actions that cannot be performed in the other elements.

See also: **WYSIWYG element, plain text element, image element, static list element, custom element form**.

Designer

One of three applications in Oracle Site Studio. Designer provides the development environment where designers can create the site and site assets, add fragments and construct contribution regions, and more. The Designer application is used with Manager and Contributor to build and maintain a site.

See also: **Manager, Contributor**.

designer

An individual who uses the Designer application to build web sites. This individual might be a web master, web developer, or member of the web team. The designer focuses on how the web site looks: the structure of the pages, the way the pages are laid out, the imagery, and the corporate identity. The actual site content is typically created and maintained by contributors.

See also: **contributor, manager, administrator**.

dynamic content

Content that opens in an area defined by a placeholder based on where the page template is used.

See also: **static content**.

dynamic list

A list of files (contributor data files or native documents) that is based on a query performed in the content server. You can have, for example, a dynamic list that queries the content server for files with the document type "Press Releases" and opens those items on the page. The list can be simple, such as a list of links, or it can be complex, such as a list showing the title, an excerpt, and a link for each file.

See also: **static list**.

editing area

Part of the Contributor editor workspace where contributors make changes to contributor data files associated with a contribution region on a Web page. Each editing area has its own toolbar and its own field for entering the data. In some cases, such as with an image or custom element, the editing area does not have any place to enter text. In many cases, there are more editing areas for data than appears on the screen. This could be because the contributor data file contains data not displayed on the web page, or that the designer has included areas meant to display only when Contributor is open.

See also: **element toolbar, contributor-only section**.

element

The smallest chunks of reusable information in an Oracle Site Studio web site. When using contributor data files on your site, you must create one or more elements in each contribution region. Each defined element is of a particular type: WYSIWYG, text only, image only, static list, dynamic list, or custom. These types characterize what the element content consists of, and, through element definitions, what editing options are available. Elements are not used with native documents, only contributor data files.

See also: **WYSIWYG element**, **plain text element**, **image element**, **static list element**, **dynamic list element**, **custom element**.

element toolbar

Each editing area in the Contributor editor has a toolbar specific to the type of area. Some toolbars have WYSIWYG items, or text-only, or image-only. Static lists and dynamic lists also have their own toolbars.

See also: **editing area**.

eXtensible Markup Language (XML)

A widely used programming language (and standard by the W3C) that facilitates the interchange of data between computer applications. XML is similar to HTML (the language used for web pages) in that both use markup tags. However, as opposed to HTML, XML tags say nothing about the presentation of the data contained in the tags, only their structure. Computer programs automatically extract data from an XML document using its associated DTD as a guide.

home page

A page that serves as an entry point to the web site. It generally contains links to the main sections of the site.

Hypertext Markup Language (HTML)

A formatting language used for documents on the World Wide Web (WWW). HTML files are plain-text files with formatting code that tells a web browser (or web-capable device) how to display text, position graphics and objects, and display links to other web pages.

HTML

See **Hypertext Markup Language (HTML)**.

HTML table

A way of presenting content in a tabular format. On a web site, HTML tables can be used to position content and therefore, design web pages. In Oracle Site Studio, you can use HTML tables to design your page templates (such as placing fragments and contribution regions in each table cell).

image element

A type of element that allows only one or more images. An image element offers options specific to adding images for a contributor (such as browsing the content server for an image, replacing an existing image, and applying a CSS class to an image).

See also: **WYSIWYG element**, **plain text element**, **static list element**, **custom element**.

JavaScript

A cross-platform scripting language that can be added to existing HTML code on a web page to create basic online functions and interactivity.

legacy site

Sites which are made with the functionality of the pre-10gR4 architecture. You can continue to use legacy sites in Oracle Site Studio 10gR4 and higher, but they do not take advantage of the new architecture and features introduced in Oracle Site Studio 10gR4 and higher.

link

An item on a web page that takes the user from one web page to another, or to a different position within a web page. Links enable visitors to navigate your web site. You can add more links yourself, usually to cross-reference another part of your site or perhaps another site.

Link wizard

A feature in Oracle Site Studio (Designer and Contributor) that enables you to quickly and easily create links to other sections, to other files, and to other sites in the content server. The multi-step wizard walks you through the necessary steps to create the link, choose a target section or file for the link, and choose a format for the link.

Manager

One of three applications in Oracle Site Studio. Manager is a web-based site management console that can be used by one or more site managers to maintain the structure of the site. Site designers can add the Manager application to a site in Designer. The look and feel and behavior can be changed for each instance of Manager.

See also: **Designer**, **Contributor**.

manager

An individual who uses the Manager application to reorganize the site navigation and hierarchy. Site managers can add or remove sections to the site without using Oracle Site Studio Designer by using a web-based tool that the site designer makes available to them. A manager typically works with one designer and multiple contributors.

metadata

"Data about data." In other words, information that describes the characteristics and properties of an item (such as title, author, content ID, and so on).

native document

A content file 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.

See also: **contributor data file**.

navigation

The way visitors make their way around a web site. The navigation on your web site is usually a collection of links that point to each main category, or section, on the site.

plain-text element

A type of element that allows only text, without modifying the font or other formatting modification. A plain-text element typically offers only the minimum

editing options for a contributor (such as cut, copy, paste, and spell checker). .

See also: **WYSIWYG element**, **image element**, **static list element**, **custom element**.

publication

The process of deploying a completed web site to a location so that it can be viewed by visitors. web sites can be deployed in Oracle Site Studio using the Oracle Site Studio Publisher. This involves gathering all of the files associated with a web site, building a static copy of the web site, and then copying the completed site to a "live" location.

region content

The data assigned to display in an editable area (contribution region) on a web page. This can be a contributor data file or a native document. Contributors are generally responsible for the region content (in the same way that designers are responsible for page templates and fragments). Once you assign region content to a contribution region, contributors can edit the content using Contributor (if using contributor data files) or a third-party application (if using native documents).

region definition

A file that defines the type of content that elements of a particular type consists of. It also specifies the content creation and switching options available to contributors for contribution regions, and sets default metadata for content files associated with these regions.

reviewer

An individual who approves content on the site. A reviewer can use the workflow options on the contribution icon to approve and reject content. A reviewer might have the sole responsibility of reviewing content, or the reviewer might be a contributor acting in both roles.

section

A category or specific area in your site hierarchy. You generally create sections in your site hierarchy that represent common parts of a web site, such as *Products*, *Services*, and *About Us*. The sections in your site hierarchy display in the navigation menus on your web site. As such, you may use the sections to organize your site, while visitors use the sections to navigate the site.

site address

A name associated with a web site that allows visitors to locate the site with a web browser.

site asset

A file associated with an Oracle Site Studio web site and used to build a web page in some way (including a fragment or region definition in the page template as well). Graphics, cascading style sheets, and custom scripts are all frequently used site assets.

Oracle Site Studio

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.

static list

A fixed number of rows and columns, each containing a contribution element (WYSIWYG, plain text, or image). Static lists can be used to bundle several elements in a table-like layout. Contributors can, therefore, add, edit, and delete the rows and columns of information, even rearrange them. Within each element, contributors have the same editing capabilities that they have in a standalone element. A static list can be implemented as an element, or as a fragment.

See also: **dynamic list**.

static list element

In Oracle Site Studio releases prior to 10gR4, static lists were created using a fragment. Starting with the 10gR4 release, static lists can be created either by using a fragment or by creating a static list element definition. The static lists function similarly, but implementing static lists as elements maximizes the reusability that all site assets have.

See also: **list fragment, plain text element, image element, static list element, dynamic list element, custom element**.

subtemplate

A partial HTML file (that is, without head and body sections) that can be inserted into a placeholder on a page template to divide the placeholder into further smaller, reusable areas with their own placeholders and contribution regions. Subtemplates may contain very simple HTML code, but they may also be quite complex, with their own scripts and the like.

Switch Content wizard

A feature in Oracle Site Studio (Designer and Contributor) that enables you to quickly and easily change the content of a contribution region. The multi-step wizard walks you through the necessary steps to change the contributor data file, native document, or even subtemplate (if the site designer has enabled changing the subtemplate).

Uniform Resource Locator (URL)

The address that defines the route to a file on an internet server (web server, FTP server, mail server, and so on).

URL

See **Uniform Resource Locator (URL)**.

validation script

A JavaScript file that defines the validation rules for element data in order to determine that the data meets the requirements (for example, it does not exceed a certain maximum length or contain some illegal characters).

web page

A file containing HTML and possibly scripts, intended for viewing in a web browser. In Oracle Site Studio, a page template is combined with contribution regions, contributor data files, native documents, images, and other site assets to form a web page that the user sees in a web browser.

web site

A collection of HTML or script-based web pages that are linked and stored on a server. In Oracle Site Studio, a web site is defined by its site hierarchy, navigation scheme, and page templates. It is stored, managed, and served up by Oracle Content Server (which

means you can take advantage of the many built-in content management features on the server).

workflow

A feature in Oracle Content Server that can be used in Oracle Site Studio to review the content of a web site before it is released for publication. A single workflow can address a portion of a single web page or all of the content on the site. Typically, multiple workflows might be used to review different portions of a web site. Designated reviewers have the authority to approve or reject content during the workflow process.

WYSIWYG element

A type of element that allows full formatting and editing of the content. A WYSIWYG element typically offers all possible formatting and editing options, such as changing the font, centering text, and creating links to new and existing files.

See also: **plain text element**, **image element**, **static list element**, **dynamic list element**, **custom element**.

XML

See **eXtensible Markup Language (XML)**.