



I N T E R W O V E N

TeamSite®
Administration Guide

Release 5.5.2L

for UNIX®

© 1999-2002 Interwoven, Inc. All rights reserved.

No part of this publication (hardcopy or electronic form) may be reproduced or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Interwoven.

Information in this manual is furnished under license by Interwoven, Inc. and may only be used in accordance with the terms of the license agreement. If this software or documentation directs you to copy materials, you must first have permission from the copyright owner of the materials to avoid violating the law which could result in damages or other remedies.

Interwoven, TeamSite, OpenDeploy and the Interwoven logo are trademarks of Interwoven, Inc., which may be registered in certain jurisdictions.

SmartContext, DataDeploy, the tagline and service mark are trademarks of Interwoven, Inc. which may be registered in certain jurisdictions. All other trademarks are owned by their respective owners. This Interwoven product utilizes third party components under the following copyrights with all rights reserved: Copyright 1995-1999, The Apache Group (www.apache.org); Copyright 1986-1993, 1998, Thomas Williams, Colin Kelley. If you are interested in using these components for other purposes, contact the appropriate vendor.



Interwoven, Inc.

803 11th Ave.

Sunnyvale, CA 94089

<http://www.interwoven.com>

Printed in the United States of America

Release 5.5.2L

Part # 80-00-10-11-00-552-300-L

Table of Contents

About This Book 11

- Notation Conventions 11
- Online Documentation Errata 13

Chapter 1: TeamSite Overview 15

- TeamSite Elements 15
 - Backing Stores 15
 - Branches 15
 - Workareas 16
 - Staging Areas 16
 - Editions 17
- TeamSite User Roles 18
 - Authors 18
 - Editors 18
 - Administrators 18
 - Masters 19
 - Other Roles 19
- TeamSite Workflow 19
 - Workflow Models 19
 - Jobs 20
 - Tasks 21
- TeamSite Architecture 21

Chapter 2: Installing TeamSite 23

- Hardware Requirements 24
 - Disk Space 24
 - CPUs 24
 - Memory 25
 - Inodes 26
 - Disk Configuration 26
 - Global Report Center Requirements 27
- Software Requirements 27
- TeamSite Client Requirements 27
 - Connecting Through the File System Interface 28
- Default File Locations 29



Installing TeamSite 5.5.2 on Solaris	30
Installing TeamSite 5.5.2 on AIX	38
Obtaining a License Key	47
Installing the License Key	47
Troubleshooting Licensing Issues	48
Upgrading to TeamSite 5.5.2	49
Configuring Web Servers	51
Specifying the Web Server Port Number	52
Specifying the Web Server HTTPD User Name	52
Configuring the iw-mount Alias	52
Configuring CGI Programs	56
Enabling Server-Side Include Requests	56
Stopping and Restarting the Web Server	59
Redirecting NSAPI HTTPS Requests	59
Configuring Samba	60
Troubleshooting Samba	63
Configuring TeamSite Clients	64
Using the Graphical User Interface	64
Using the File System Interface	65
Loading Content	69
Creating a Subbranch	70
Creating a Workarea	73
Populating an Initial Workarea	75
Submitting Files to the Staging Area	76
Publishing a New Edition	78
Uninstalling TeamSite	79
The iwui User	80

Chapter 3: Managing Access 81

Security	81
Users	83
TeamSite Roles Overview	83
Adding and Removing Users	84
Adding Users	84
Deleting Users	85
Access Control	86
Group Membership	86
Checking User Roles	89

Locking Models	90
Submit Locking	90
Write Locking	91
Permissions	91

Chapter 4: Using the Interwoven Administration GUI 99

Navigation	101
Apply, Refresh, and Cancel	101
Logging In To the Interwoven Administration GUI	102
Viewing System Information	102
Editing Roles	104
Setting Host Permissions	106
Setting TeamSite Permissions	107
Configuring General TeamSite GUI Preferences	109
Changing Area Labels in the TeamSite GUI	111
Configuring the General Proxy Settings	111
Configuring Proxy Mappings	112
Configuring Server Performance	114
Configuring TeamSite Log Files	116
Viewing TeamSite Log Files	117
Performing Server Operations	119
Abort	119
Freeze or Unfreeze	119
Reset	120

Chapter 5: Configuring the TeamSite Server 121

Configuring GUI Appearance	124
Configuring TeamSite Area Labels	124
Configuring Edition Views	126
Configuring History Views	127
User Profiles	127
Configuring GUI Functionality	128
Disabling Editor Publish Capability	128
Enabling and Disabling SmartContext Editing	128
The Casual Contributor Interface: Adding Editing and Task Links to Web Pages	129
Setting the Default LaunchPad Interface	131
Setting Unique Server Names for LaunchPad to Recognize	132
Setting Login Authentication Expiration	132



Configuring Preview Windows	133
Custom Menu Items	134
Configuring Submit Button Behavior	139
Disabling Menu Items	140
Disabling Directory Operations	142
Disabling Unlocked File Auto-Upload	143
Setting the Number of Jobs Listed in the To Do List	143
Configuring Job Attribute Filters and Settings	144
Configuring Email Settings	145
Configuring Server Functionality	146
Specifying the Encoding of the iw.cfg File	146
User and Role Authentication	146
Webserver UID	152
Web Daemon	152
Servlet Engine	152
Main Branch Settings	153
Locked File Submission	154
Submit and Update Logs	154
Branch and Workarea Security	154
Default Permissions	155
Group Remapping	156
File Locations	156
Autoprivate	158
New File Templates	161
Launching Files Through iProxy	163
Configuring the TeamSite Server Locale	163
Configuring Server Performance	164
Cache Size	164
RPC Threadcount	165
File System Threadcount	165
Filesystem Active Area Cache	166
Throughput Monitors	166
Detecting Low Disk Space and Inode Count	167
Submit Filtering	167
RCS Macro Expansion	172
Configuring the TeamSite Web Daemon and Proxy Server	175
About the TeamSite Web Daemon	175
About the Proxy Server	175

Applying Changes to Proxy Configuration	177
Configuring TeamSite Web Daemon and Proxy Server Operation	177
Resolving Relative and Absolute Paths	178
Resolving Fully-Qualified URLs	182
Redirecting TeamSite Views to Different Areas	186
Configuring TeamSite to Use Different Web Servers	189
Configuring External Remappings	190
Host Header Remappings	191
Configuring SSI Remapping	192
Configuring Proxy Failover	192
Debugging Your Proxy Server Configuration	194
TeamSite Embedded Failsafe	195

Chapter 6: Configuring Metadata Capture and Search 197

Metadata Capture	197
Overview	198
Components	198
Configuring Metadata Capture	200
Metadata Capture End Result	221
Metadata Capture and TeamSite Workflow	222
Metadata Search	223
Overview	223
Prerequisites	223
Components	224
Configuring Metadata Search	226

Chapter 7: Managing the TeamSite Server 229

Checking Server Status	230
Verifying Server Operation	230
Checking for Multiple Servers	230
Checking Request Handling	231
Verifying the Server Mount	231
Finding the Installation Directory	232
Reviewing TeamSite Logs	233
Monitoring the Server Load	234
Starting and Stopping the Server	234
Managing the OpenAPI Server	234
Verifying that the OpenAPI Server is Running	234



Starting and Stopping OpenAPI	235
Reconfiguring iwwebd to Recognize a New IP Address	235
Re-Encrypting User Authentication Information	235
Troubleshooting	236
Repairing the Backing Store	236
Managing Server Resources	242
Disk Space	242

Chapter 8: TeamSite Backing Stores 247

Backing Store Overview	247
Planning the Backing Store Conversion	248
Conversion Overview	249
Conversion Prerequisites and Tips	251
Converting Backing Stores Using the GUI	252
Converting Backing Stores from the Command Line	256
iwconvert Command-Line Tool	256
Conversion Procedure	259
Creating Multiple Backing Stores	261
Defining Backing Stores in the iw.cfg File	262
Creating Backing Stores Using the iwstoreadm CLT	265
Administration CLTs	267
iwstoreadm	267
iwidmap	268
iwmigrate	270
iwconvertserver	271
iwcpfile	271
iwcpwa	272
iwwfconvert	273
UID Changes to the TeamSite Backing Store	274

Chapter 9: Backing Up TeamSite 275

Integrating with Third-Party Backup Solutions	275
Suggested Strategies for Incremental Backups	277

Appendix A: TeamSite Configuration Files 279

Location of iw.cfg	281
Location of Roles Files	281

Appendix B: Specifying Content Encoding 283

regex_map	Defined	284
Simple regex_map Example	285	
The regex_map Format	286	
Rule Syntax	286	
Regular Expression Syntax	287	
Variables	287	
Application Variables	288	
Intermediate Variables	288	
Interpolation of Variables and Captured Subexpressions	289	
Quoting	292	
Strategies for Effective regex_maps	294	
Internationalization and regex_map	296	
SmartContext Editing and file_encoding.cfg	296	
Source Differencing and Merging and file_encoding.cfg	297	
Sample file_encoding.cfg	298	
Advanced regex_map Example	299	

Appendix C: High Availability TeamSite 301

HA Watchdog	301	
About HA Watchdog	301	
TeamSite HA Watchdog Components and Processes	302	
Installing TeamSite HA Watchdog	303	
Configuring TeamSite HA Watchdog	304	
Starting and Stopping the Server Under HA Watchdog	306	
Uninstalling TeamSite HA Watchdog	307	
Related Documentation	307	
HA Hot Standby	307	
About HA Hot Standby	307	
Installing TeamSite and High Availability Hot Standby	309	
Applying the TeamSite-Sun Cluster Integration Patch	311	
Verifying the hainterwoven Data Service Failover	312	
Additional Information	313	

Appendix D: Internationalization 315

Supported Client and Server Platforms	315	
Servers	316	
Clients	316	



Browsers	316
Supported TeamSite Server Locales	317
Supported Content	317
Localization Overview	317
What's Been Translated?	318
What's Not Been Translated?	318
Limitations and Assumptions	319
Backing Stores and Character Encoding	320
About UTF-8	320
CCI URLs with Multibyte Characters	321
Interfacing with Localized Operating Systems	321
Accessing the Localized Interface	322
CLT Internationalization	322
CGI Internationalization	322
Specifying File Encoding of Text Files	322
Text Editor Encodings	324
Behavior of Netscape Navigator	324
Configuring Netscape for Multibyte Characters	325
Usage Scenarios	325

Appendix E: Client/Server Compatability 329

Appendix F: Integrating with SiteMinder 337

Requirements	337
Authentication Overview	338
Integration Procedure Overview	339
Creating a TeamSite Server URL File	340
Copying the iwtssmar.so File	341
Configuring SiteMinder	341
Configuring the Reverse Proxy	343
Integrating Using the Apache Reverse Proxy Web Agent	344
Integrating Using NSSRP	344

Appendix G: Root Access to TeamSite 349

Index 353

About This Book

The *TeamSite Administration Guide* is a guide to installing, configuring, and maintaining TeamSite. It is primarily intended for TeamSite Administrators and Master users, web server administrators, and system administrators. Users of this manual should be familiar with basic UNIX commands and be able to use an editor such as emacs or vi.

Many of the operations described in this manual require root access to the TeamSite server, these operations are listed in Appendix G. If you do not have root access to the TeamSite server, consult your UNIX system administrator.

It is also very helpful to be familiar with regular expression syntax. If you are not familiar with regular expressions, it is recommended that you consult a reference manual such as *Mastering Regular Expressions*, by Jeffrey Friedl.

Some TeamSite configuration files make use of XML. For more information about XML, consult a reference manual or the online specification at <http://www.xml.com/axml/testaxml.htm>.

Notation Conventions

This manual uses the following notation conventions:

Convention	Definition and Usage
Bold	Text that appears in a GUI element (for example, a menu item, button, or element of a dialog box) and command names are shown in bold. For example: Click Edit File in the Button Bar.
<i>Italic</i>	Book titles appear in italics. Terms are italicized the first time they are introduced. Important information may be italicized for emphasis.

Convention	Definition and Usage
Monospaced	<p>Commands, command-line output, and file names are in monospaced type. For example:</p> <p>The <code>iwextattr</code> command-line tool allows you to set and look up extended attributes on a file.</p>
<i>Monospaced italic</i>	<p>Monospaced italics are used for command-line variables. For example:</p> <pre>iwckrole <i>role user</i></pre> <p>means that you must insert the values of <i>role</i> and <i>user</i> yourself.</p>
Monospaced bold	<p>Monospaced bold represents user input. The % character that appears before a line of user input represents the command prompt and should not be typed. Your system may not use this command prompt. For example:</p> <pre>% iwextattr -s project=proj1 //IWSERVER/default/main/dev/WORKAREA/andre/products/index.html</pre>
<i>Monospaced bold italic</i>	<p>Monospaced bold italic text is used to indicate a variable in user input. For example:</p> <pre>% iwextattr -s project=projectname workareavpath</pre> <p>means that you must insert the values of <i>projectname</i> and <i>workareavpath</i> when you enter this command.</p>
[]	Square brackets surrounding a command-line argument mean that the argument is optional.
	Vertical bars separating command-line arguments mean that only one of the arguments can be used.

Online Documentation Errata

Additions and corrections to this document are available in PDF format at the following Web site: <http://support.interwoven.com>

When you reach this site:

1. Click **Download**.
2. Enter your user name and password.
3. Click **All Documentation**.
4. Click **Current Release Notes**.
5. Click the link to the appropriate PDF file.

Chapter 1

TeamSite Overview

This chapter introduces the following three major TeamSite concepts and concludes with a description of the TeamSite system architecture:

- TeamSite Elements
- TeamSite User Roles
- TeamSite Workflow

TeamSite Elements

Backing Stores

The backing store is a large directory created by the TeamSite installation program that contains TeamSite files and metadata. By default, the backing store is located in `/local/iw-store`.

Previous releases of TeamSite have been limited to one backing store per TeamSite server. This release supports as many as eight backing stores per TeamSite server (the first created automatically by the installation program, and the others created by the TeamSite administrator). The functionality that enables multiple backing stores is known as *MultiStore*.

For detailed information about backing stores, MultiStore, and converting your existing backing store (upgrade customers only), refer to Chapter 8 and the *Backing Store Conversion Guide* (available in PDF format only).

Branches

TeamSite provides *branches* for different paths of development for a Web site. Branches can be related to each other (for example, alternate language versions of the same Web site) or they may be completely independent. Each branch contains all the content for a Web site.

A single branch contains archived copies of the Web site as *editions*, a *staging area* for content integration, and individual *workareas* where users may develop content without disturbing one another. Branches can also contain *sub-branches*, so that teams may keep alternate paths of development separate from each other. Content can be easily shared and synchronized across branches and sub-branches. Users may work on one branch or on several, and the number of branches on a system is not limited.

Branches facilitate distributed workflow because they allow separate teams to work independently on different projects. Because all branches are located on the same TeamSite server, it is easy for one team to incorporate the work of another into their project.

Workareas

Each *workarea* contains a virtual copy of the entire Web site, which may be modified in any way without affecting the work of other contributors. Users who have access to a workarea may modify files within that workarea and view their changes within the context of the entire Web site before integrating their work with that of other contributors. Users can lock files in each workarea, eliminating the possibility of conflicting edits.

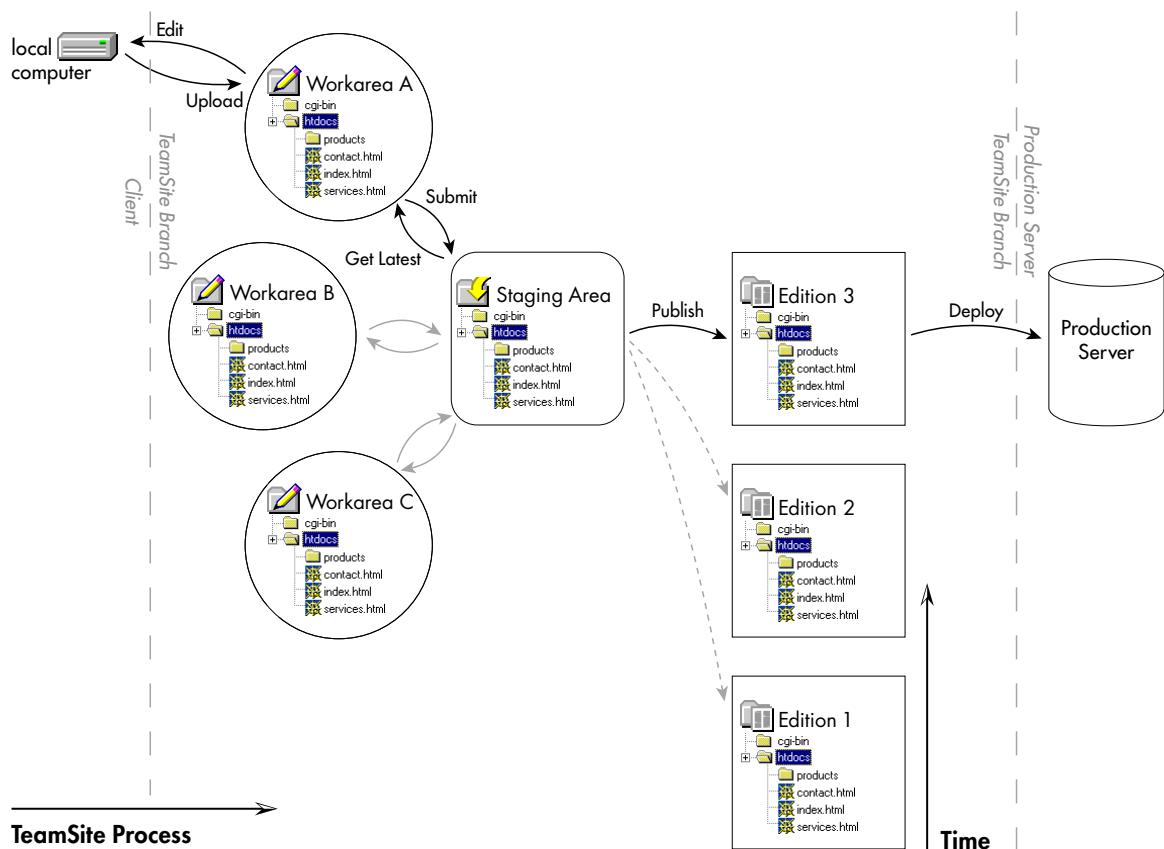
All changes that are made to files in a workarea are kept completely separate from other workareas and the staging area until the user chooses to promote his changes to the staging area. Within a workarea, users may add, edit, or delete files, or revert to older versions of files without affecting other users.

Staging Areas

Each branch contains one *staging area* where contributors incorporate their changes with the work of others. Users submit files from their workareas to the staging area to integrate their work with other contributions, and test the integrity of the resulting Web site. Because the staging area is an integrated component of the system, conflicts are easily identified and different versions of the same file can be merged, rather than overwritten.

Editions

Editions are read-only snapshots of the entire Web site, taken at sequential points in its development. Contributors can create new editions any time they feel their work is well integrated, or any time they want to create an update to the Web site for reference or deployment. Each edition is a fully functional version of the Web site, so that users may see the development of the Web site over time and compare it with current work.



TeamSite branches contain private workareas, which contain complete virtual copies of the Web site; staging areas, where contributors integrate their work; and editions, which are read-only snapshots of the Web site at various points in its development. Each area contains a virtual copy of the entire Web site. Content is submitted from workareas to the staging area, and the staging area is then published as an edition. Older editions are available for reference.

TeamSite User Roles

Authors

Authors are primary content creators. All work done by Authors goes through an explicit approval step. They can receive assignments from Editors, which are displayed in To-Do lists when Authors log in to TeamSite. Authors can access TeamSite from a simple browser-based interface, and do not need to be sophisticated computer users.

In order to test and QA their work, Authors have full access to the content in their Editors' workareas, but do not need to concern themselves with the larger structure and functionality of TeamSite. The Author role is appropriate for non-technical users, or for more technical contributors who do not need access to TeamSite's extended functionality, such as TeamSite's advanced version management features.

Editors

Editors own workareas. They create and edit content, just as Authors do, but they are primarily responsible for managing the development taking place within their workareas. This includes assigning files to Authors and submitting completed content to the staging area, and it may include publishing editions.

Editors have access to specialized TeamSite content and workflow management functions. Editors are generally "managerial" users, who primarily supervise the work of Authors, or self-managing "power" users, who need TeamSite's extended functionality to manage their own content.

Administrators

Administrators own branches. They have all the abilities of Editors, but they are primarily responsible for the content and functioning of their branch. Administrators can manage project workflow by creating new workareas for Editors and groups, and by creating sub-branches of their own branch to explore separate paths of development.

An Administrator is the supervisor of the project being developed on his branch. He may be the webmaster for a particular version of the Web site, or a project manager.

Masters

Master users own the Web site. They can perform all the functions of Editors and Administrators on any branch. The Master user owns the main branch, from which all sub-branches are created. The Master user is generally involved in the installation of TeamSite, and can reconfigure TeamSite on a system-wide basis.

Other Roles

The TeamSite installation program also creates the following roles and corresponding uid files:

- content-provider—Provides access to Open Syndicate contributor functionality?
- od-admin—Provides access to Open Deploy administrator functionality.
- od-user —Provides access to Open Deploy end-users functionality.
- syndicate-admin—Provides access to Open Syndicate administrator functionality.

These roles are used by other Interwoven products and are described in their corresponding user and administrator documentation.

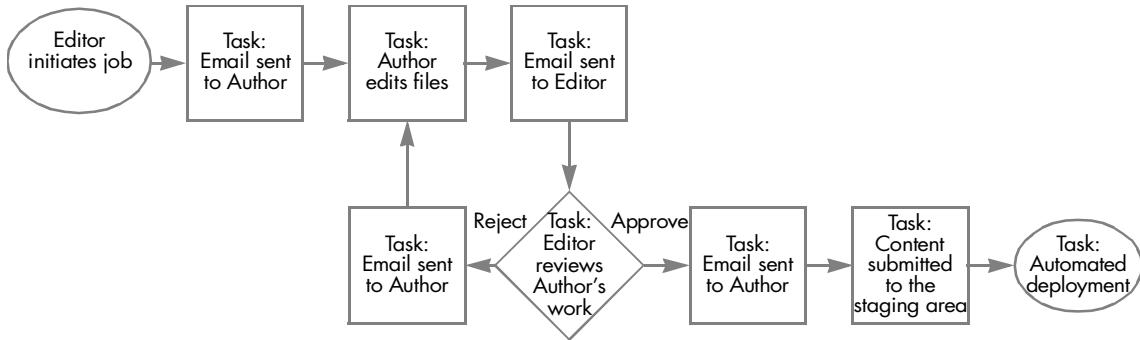
TeamSite Workflow

Workflow Models

A *workflow model* is a general workflow configuration that can be used repeatedly. Each workflow model describes a process which may include user tasks and a wide variety of automated tasks. Workflow models are configured by the system administrator or by the Interwoven Client Services organization.

For more information about configuring different workflow models, consult the *TeamSite Workflow Developer's Guide*.

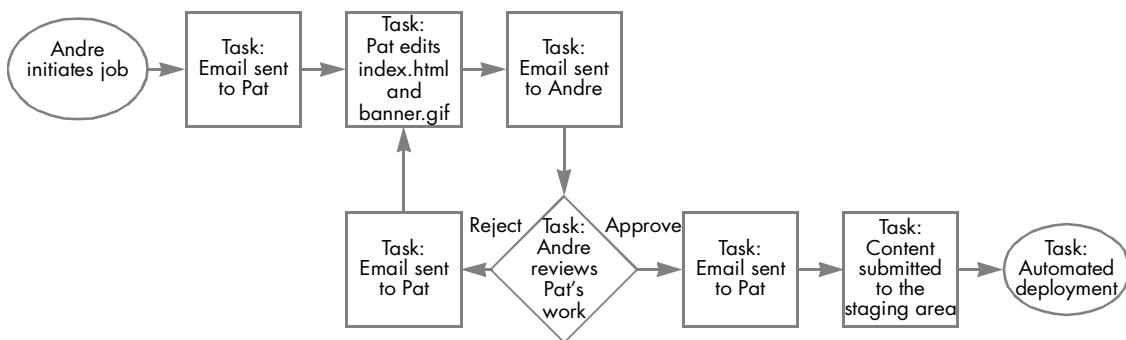
Below is a diagram of a very simple assign-edit-approve workflow model. Email is sent to the participants at every stage of the process, and some automated tasks are performed at the end.



Jobs

A *job* is a set of interdependent tasks. One example of a TeamSite job would be the set of tasks needed to prepare a new section in a marketing Web site to support a new product launch.

Each job is a specific instance of a workflow model. When a job is created, the job creator must supply all the specific information for that job. For example, the workflow model above might be used to create the job below.



Because jobs follow predefined workflow models, tasks cannot be added to or removed from individual jobs.

Tasks

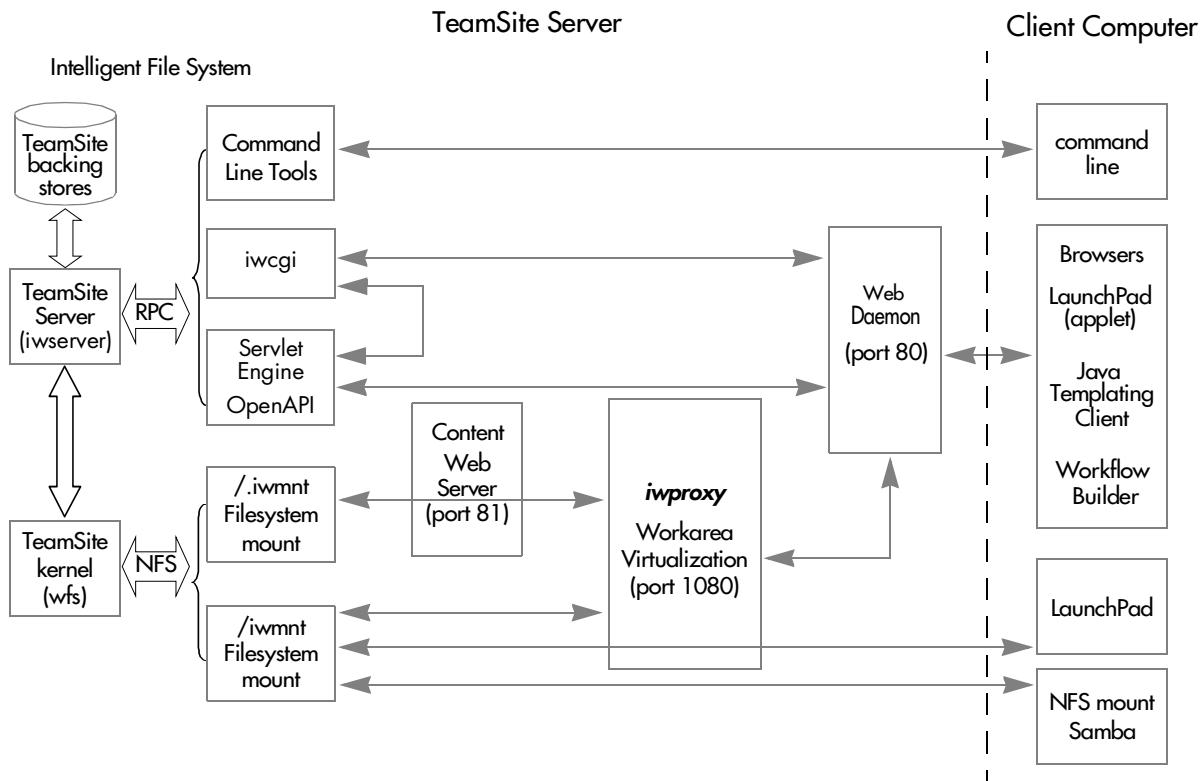
A *task* is a unit of work performed by a single user or process. Each task in a job is associated with a particular TeamSite workarea and carries a set of files with it. The user or process owning a task can modify, add files to, or remove files from the task.

Tasks have two possible states: active and inactive. A task becomes active when its predecessor tasks signal it to do so (predecessor tasks and conditions for activation are all configured as part of the workflow model). Once the task has been activated, users or external programs can work on it. For example, once a user task has been activated, the user can work on the files contained in the task. Once an external task has been activated, the appropriate external program can run on the files contained in the task. Inactive tasks are tasks that have been completed, or that have not been activated yet.

TeamSite Architecture

TeamSite's Intelligent File System (IFS) is composed of the TeamSite server and kernel, the TeamSite backing store of files and metadata, a suite of command-line tools, TeamSite CGI, proxy servers for access through the TeamSite browser-based GUI, and file system mounts for access through the file system interface.

The Intelligent File System is the core of the TeamSite system, where detailed information about the Web site, the Web assets, Web asset metadata, the production process and the users is stored. The Intelligent File System collects and maintains metadata on TeamSite files, directories, and areas, and allows TeamSite to process and present information according to who is asking for the information, and under what conditions. By using an object oriented design within a file system architecture, TeamSite combines extensive metadata tagging with open access and file system performance for Web content.



The client computer connects to the TeamSite server in several ways. Requests from the browsers or LaunchPad are routed through the TeamSite Web daemon, which allows consistent views of TeamSite areas. The double proxy server redirects hard-coded links within the Web site. Requests through the file system interface (NFS mount / Samba) and command-line tools, which do not go through the webserver, are not routed through a proxy server.

Chapter 2

Installing TeamSite

This chapter describes the process for installing, upgrading, and licensing TeamSite 5.5.2 and configuring all related system resources. The actual configuration of TeamSite is described in Chapters 3 through 6. The following topics are included in this chapter:

- Hardware Requirements
- Software Requirements
- TeamSite Client Requirements
- Default File Locations
- Installing TeamSite 5.5.2 on Solaris
- Installing TeamSite 5.5.2 on AIX
- Upgrading to TeamSite 5.5.2
- Configuring Web Servers
- Configuring Samba
- Configuring TeamSite Clients
- Loading Content
- Uninstalling TeamSite
- The iwui User

Refer to the *TeamSite Templating Developer's Guide* for information about installing TeamSite Templating.

Hardware Requirements

Before beginning the TeamSite installation, ensure that your system includes the hardware described in the following sections.

Disk Space

Your system must have at least 800 MB of disk space if you are installing on Solaris, and 1.3 GB of disk space if you are installing on AIX, for the TeamSite program files, plus an additional five to 10 times the total amount of disk space you expect the Web site content files to consume.

The *Backing store* is a large directory that contains TeamSite files and metadata. Ensure that the location of the backing store has room for at least three times the size of the entire Web site (for example, if the Web site will be 200MB in size, the backing store should have at least 600MB of disk space available). For ease of maintenance, the backing store should be installed to its own partition.

CPUs

The following CPU recommendations are based on the different types of users who will be concurrently using TeamSite. Note that some types of users tend to do CPU-intensive operations (including Get Latest, Submit, or Compare) while other users tend to do more lightweight operations such as editing files and browsing directories.

All CPUs should be at least 400 MHz, and on AIX, must be capable of running a 64-bit kernel.

To determine the number of CPUs you need:

1. Determine how many of your users will be using TeamSite intensively (for example, members of a Web development team), moderately, or mildly (such as occasional contributors to the company intranet).
2. Determine how many of each type of user will be using TeamSite concurrently.
3. Locate the number of concurrent users in the following table to determine the number of CPUs you need for each type of user, and add them together to get your total number of CPUs.

Note: You must have at least two CPUs.

		Concurrent Users		
		Intense	Moderate	Mild
CPUs	1	25	50	100
	2	50	100	200
	4	100	200	400
	8*	200	400	800

* If the number of concurrent users is greater than the numbers in this row, please contact Interwoven Client Services.

For example, if you have a total of 60 intense, 250 moderate, and 2000 mild users, and you expect 40% of the intense users, 20% of the moderate users, and 10% of the mild users will be using TeamSite concurrently, then your concurrent users and total number of required CPUs would be as follows:

$$\begin{aligned}
 60 \text{ intense users} * .4 &= 24 \text{ concurrent intense users} = 1 \text{ CPU} \\
 250 \text{ moderate users} * .2 &= 50 \text{ concurrent moderate users} = 1 \text{ CPU} \\
 2000 \text{ mild users} * .1 &= 200 \text{ concurrent mild users} = \underline{\underline{2 \text{ CPUs}}} \\
 &\qquad\qquad\qquad \text{Total} \qquad\qquad\qquad 4 \text{ CPUs}
 \end{aligned}$$

Memory

In general, TeamSite requires 1 GB of memory for each CPU (see above). To calculate memory requirements more precisely, use the following formula:

$$(1 \text{ GB}) + (\text{cache size setting} * 4 \text{ KB}) + (\text{total number of concurrent users} * 4.6 \text{ MB})$$

where the cache size setting is specified using the `cachesize` parameter in `iw.cfg`. This setting is 30,000 by default. For information about changing this setting, see “Cache Size” on page 164.

For example, the memory requirements for the example system specified above (which has 24 + 50 + 200 concurrent users), with the default cache size setting, would be:

$$1 \text{ GB} + (30000 * 4 \text{ KB}) + (4.6 \text{ MB} * (24 + 50 + 200)) = 2.38 \text{ GB}$$

If you encounter a significant amount of memory swapping, you should either increase the `cachesize` setting in `iw.cfg` or install more memory.

Inodes

To determine how many inodes your server will require, use the following formula:

$$\# \text{ inodes} = (\# \text{ branches})(\# \text{ average files in staging area per branch})(\# \text{ average historical versions/file}) (1 + (\% \text{ of files having extended attributes})/100))(\text{safety-factor})/100$$

For example, if your TeamSite server has three branches, with 20,000 files in the staging area of each branch, (on average), ten versions of each file in its history list (on average), seven percent of files have extended attributes, and you want to use a 1.5x safety factor:

$$\begin{aligned}\# \text{ inodes} &= 3 * 20,000 * 10 * (1 + .07) * 1.5/100 \\ &= 9630 \text{ inodes}\end{aligned}$$

Disk Configuration

For maximum disk space efficiency, the TeamSite backing store should be installed on drives formatted with a 512 byte block or fragment size. For ease of maintenance, you may want to install the backing store on its own partition.

It is recommended that you use RAID 0+1 to configure your environment. RAID 5 can also be used for environments with a relatively low number of writes as a percentage of total accesses. Because TeamSite environments generally have a large percentage of writes, RAID 0+1 should provide better overall performance.

In addition to using RAID configurations, it is recommended that you use the fastest available SCSI controllers (160 MB/Sec transfer rate) and SCSI drives (10,000 RPM).

Note: Software RAID solutions are not recommended because they are very CPU-intensive.

Global Report Center Requirements

The TeamSite Global Report Center is an optional installation that requires an additional 25 MB of disk space, plus 10-50 MB for data storage. The Global Report Center also requires approximately 5 MB of physical memory. The OpenDeploy Global Report Center has the same requirement. Therefore, you should plan on approximately 10 MB of physical memory for the Global Report Center if you install both TeamSite and OpenDeploy.

Software Requirements

TeamSite runs on the same system as your Web site development server. It is recommended that you configure your Web site development server as a dedicated server. It should not run applications other than the Web server software and TeamSite.

The following software is required or recommended to run TeamSite on US operating systems (refer to Appendix D, “Internationalization” for details about other operating systems):

- Operating system—For Solaris, Sun Solaris 2.6 or 32- or 64-bit System 7 or System 8 (all require patches for JDK 1.3). Check <http://sunsolve.sun.com> for the relevant patch clusters for the various Solaris versions. For AIX, AIX 5.1 with update 5100-02_AIX_ML and patch IY28762. The AIX system must be running a 64-bit kernel.
- Web server software—Netscape Enterprise Server 3.0, iPlanet 4.1, Apache 1.3.9, 1.3.12, or 1.3.19. AIX supports Apache 1.3.26 or later.

Note: TeamSite for Solaris must be installed on an UltraSparc platform.

TeamSite Client Requirements

End-users access TeamSite through browser-based thin-client technology. The only hardware requirements for client systems are that the RAM, CPU, local storage, and networking capability must be sufficient to operate a web browser and the editing applications of the user’s choice. TeamSite’s thin-client interface does not require you to install any other client software unless you will be editing files through the TeamSite GUI.

Not all TeamSite features are compatible with all browsers on all client platforms. The following table shows compatibility for Netscape and Internet Explorer:

Client	Netscape	Internet Explorer*
Windows 98, Windows NT, Windows 2000	4.7x	5.0-5.5, 6.0
UNIX	4.7x	Not supported
MacOS	Not supported	5.1

- * Some versions of Internet Explorer do not include the Java Virtual Machine. If you do not have the Java Virtual Machine, you can download it from Microsoft's Web site at www.microsoft.com.

If you are using Netscape browsers to display multi-byte characters, you must select **Edit > Preferences > Appearance > Fonts** and set the **Use my fonts, overriding page-specified fonts** option.

Connecting Through the File System Interface

To connect to TeamSite using the file system interface, users must have a network connection and the ability to interact with a networked file system (for example, FTP, NFS, PCNFS, Samba server, FTP client, Windows networking). Before installing and configuring any of these protocols, you should be familiar enough with them to perform basic configuration and startup procedures. For more information, see “Using the File System Interface” on page 65.

Default File Locations

By default, TeamSite is installed in the following locations (you may select alternate locations for some of these files during the installation process):

Default Directory	Contents
/usr/iw-home	Default location of TeamSite program files. The location of this directory may be changed during installation or when the server is stopped.
/local/iw-store	<p>Default location of the TeamSite backing store (this is where TeamSite stores files and metadata for workareas and editions). This directory can consume large amounts of disk space. The location of this directory may be changed during installation or when the server is stopped. To find where this directory is located, use the command-line tool <code>iwgetstore</code> (see <i>TeamSite Command-Line Tools</i>).</p> <p>Note: The contents of this directory should never be edited by hand in any way. Tampering with this directory can irreparably corrupt the data stored in TeamSite.</p>
/iwservr	Local file system mount projection directory. Clients mount to this directory to access Web site data. No actual data is stored in this directory, and its location cannot be changed.
/iwmnt	NFS server mount point. This directory is used to access Web site data when working directly from the server. The location of this directory can be changed; however, Web server aliases must be updated to reflect this.
/.iwmnt	NFS server mount point. This is a noncaching alias used by the Web server. The location of this directory can be changed; however, the Web server alias must be updated to reflect this change.

Installing TeamSite 5.5.2 on Solaris

This section describes the TeamSite installation process. Ensure that you have satisfied the hardware and software requirements described in the preceding sections before starting the installation program. The installation process is recorded in the `iwinstall.log` file in `iw-home/install/`.

If you are upgrading to TeamSite 5.5.2, proceed to “Upgrading to TeamSite 5.5.2” on page 49.

To install TeamSite on your Web site development server:

1. Log in as `root` on the system where you want to install the TeamSite server.

Note: Do not use `sudo su -` or `su` to gain root access when installing TeamSite. Using `sudo su -` or `su` instead of `su -` may cause problems with shutdown, startup, or installation operations.

2. Insert the TeamSite CD-ROM and browse to the top level directory.
3. Copy the installation file, `IWOV-sol.5.5.2.BuildNumber.tar.gz`, to the directory in which you want the TeamSite application to reside.
4. Uncompress and expand the `IWOV-sol.5.5.2.BuildNumber.tar.gz` file by running the following command:
`% gunzip -c IWOV-sol.5.5.2.BuildNumber.tar.gz | (cd /parent_directory; tar xvzf -)`

The default `parent_directory` for TeamSite is `/usr`. This command creates a directory called `iw-home` under the `parent_directory`. The `iw-home` directory contains the TeamSite program files.

5. Locate the TeamSite installation file `iwinstall` in the subdirectory `iw-home/install`. From this directory, run the following command to starts the TeamSite installation process:
`% ./iwinstall`

The following message and prompt is displayed:

```
Starting Installation of Interwoven TeamSite (tm)
iwininstall version <5.5.2 Build 6011 SYM Interwoven 20020208>
Start Time: Fri Feb 8 04:20:00 PDT 2002

This product contains portions of code under the following copyrights with
all rights reserved: Copyright 1997 Eric Young; Copyright 1995-1999, The
Apache Group; Copyright 1999, ExoLab Group; Copyright 2001, JavaServer Pages,
Hans Bergsten (http://TheJSPBook.com/).
```

```
Do you agree with the license agreement? { y or n , default n }:
```

6. Type **y** and then press **Return** to accept the license agreement.

The following message and prompt is displayed:

```
This version of TeamSite requires OS patches for JDK1.3 installed. If this
machine does not have the OS patches installed, you should download proper
patches from this site "http://java.sun.com/j2se/1.3/install-solaris-patches.html". Then, install the patches before you install the Interwoven
TeamSite.
```

```
Did you have the patches installed already? { y or n , default n }:
```

7. Ensure that the operating system patches are installed, type **y** and then press **Return**.

The installation program checks to ensure there is sufficient disk space for TeamSite:

```
TeamSite Installation needs at least 300MB of free disk space.
Current disk space on this machine is: 2240MB
Do you wish to continue?[y]:
```

8. Press **Return** to accept the default, or type **y**.

The following message and prompt is displayed:

```
Proceeding to Install with enough disk space.
Copying TeamSite shared libraries to /opt.
```

```
Creating Interwoven TeamSite UI Daemons User (iwui)...
```

```
Please choose a UID for the new user, or press <Enter> to accept the
next available UID:
```

9. Press **Return** to accept the default, or type **n** at the prompt to specify a different UID.

Note: For detailed information about the UI daemons user (**iwui**), see page 80.

The installation program confirms the creation of the `iwui` user, copies the TeamSite program files and creates the TeamSite directory:

```
The user "iwui" has been created successfully.
```

```
STEP 0: Copying platform specific files.  
Interactively installing...
```

```
TeamSite server and related processes are stopped
```

```
STEP 1: Install default IW_HOME directory.  
Creating new /etc/defaultiwhome ...  
IW_HOME is now /iw-home  
Installing /usr/bin/iwgethomed  
Moving obsolete file/directory to /iw-home/OBSOLETE/iwdeploy
```

```
STEP 2: Install /kernel/fs/wfs
```

```
STEP 3: Install /etc/name_to_sysnum.  
wfs (181) has been added to /etc/name_to_sysnum
```

```
STEP 4: Install and export /iwserv  
/iwserv nfs share entry has been added to /etc/dfs/dfstab
```

```
STEP 5: Install software in /etc ...  
Installing items in /etc/init.d...  
Copying /etc/iw.cfg to /etc/iw.cfg.bak  
Installing iwserv to start at boot time: /etc/rc3.d/S16iwserv  
Installing iwserv to stop at shutdown time: /etc/rc2.d/K16iwserv  
Installing iw.local to start at boot time: /etc/rc3.d/S99iw.local  
Installing iw.local to stop at shutdown time: /etc/rc3.d/K99iw.local  
Installing iw.reboot to start at boot time: /etc/rc3.d/S15iwserv.reboot
```

You are prompted to specify locations for TeamSite's log files, mount point, and backing store.

```
STEP 6: Install default configurations in /etc/defaultiwhome... files  
Enter the location to store all the log files:(default:/var/adm)
```

```
iwevents.log, iwtrace.log ,iwserv.log and other logs are located in  
/var/adm Accept default mount directory location /iwmnt? [y] y
```

```
Default entry /iwmnt saved in file /etc/defaultiwmount  
Accept default backing store directory location /local/iw-store ? [y] y
```

The *backing store* is a large directory that contains TeamSite files and metadata. Ensure that the location of the backing store has room for at least three times the size of the entire Web site (for example, if the Web site will be 200MB in size, the backing store should have at least

600MB of disk space available). For ease of maintenance, the backing store be installed to its own partition.

10. Press **Return** to accept each of the default locations, or type **n** at the prompt to specify a different location.

Note: If you specify an alternate location for the backing store, you must use ASCII characters.

The system confirms the creation of the log files and the backing store and writes that location of the backing store to the /etc/defaultiwstore file:

```
Default entry /local/iw-store saved in file /etc/defaultiwstore
```

This file tells TeamSite where the backing store is located.

The installation program continues by installing the Perl programs:

```
STEP 7: Install Perl programs.  
Installing /iw-home/iw-perl  
It will take a few minutes...  
Creating /iw-home/local/config/wft/available_templates.ipl  
Creating /iw-home/local/config/wft/available_templates.cfg  
Creating /iw-home/events subsystem/conf/jmsconfig.xml  
Creating /iw-home/events subsystem/conf/events subsystem.properties
```

The installation program continues to create and populate access control groups:

```
STEP 8: Create group members for access control  
The following files will contain group membership information for  
Administrators, Editors and Authors. YOU MAY MODIFY THEM BEFORE OR  
WHILE Interwoven TeamSite IS RUNNING BY EDITING THE FILES DIRECTLY.
```

```
/iw-home/conf/roles/master.uid  
/iw-home/conf/roles/admin.uid  
/iw-home/conf/roles/editor.uid  
/iw-home/conf/roles/author.uid  
/iw-home/conf/roles/od-admin.uid  
/iw-home/conf/roles/od-user.uid  
/iw-home/conf/roles/content-provider.uid  
/iw-home/conf/roles/syndication-admin.uid
```

```
Please select your option for initial <master> group setup  
[1] Assign everyone in my /etc/passwd file to be an master.  
[2] I'd like to manually assign a few members now.
```

```
Please enter your choice: [1]
```

11. Press **Return** to accept the default and automatically assign all members listed in the password file to the specified TeamSite role (master), or type **2** and manually specify individual users for each role.

When manually assigning user names for a role, type each name separated by a space, then press **Return** when you are finished.

You can modify these files at any time (see page 84). Deleting a user name from a file removes TeamSite access privileges for the role. Adding a user name to a file grants access privileges for the role.

The system adds the specified users to the `master.uid` role file and prompts you to specify users for the `admin.uid`.

```
Installing master roles in /iw-home/conf/roles
Done creating roles in /iw-home/conf/roles
```

```
Please select your option for initial <admin> group setup
[1] Assign everyone in my /etc/passwd file to be an admin.
[2] I'd like to manually assign a few members now.
```

Please enter your choice: [1]

12. Repeat step 11 for each of the remain role files.

After the final group is created, the following prompt is displayed:

```
STEP 9: Install Interwoven icons and cgi programs for NCSA or Apache
servers. Are you using either an NCSA or an Apache Web server? [y]
```

13. If you are using an Apache or NCSA Web server, press **Return** or type **y** to can install a series of aliases that enable you to access TeamSite directly from within the Web server.

If you do not want TeamSite to install these aliases, type **n**. If you are using a Netscape or iPlanet Web server, type **n** (continue with step 15).

If you select to create the aliases, the installation program displays the following prompt:

```
Is "/usr/local/etc/httpd" your server httpd location? [y]
```

14. Press **Return** to accept the default, or type **n** and enter the Web server's `httpd` location (if you type **n**, you will need to configure your Web server manually after the TeamSite installation is finished as described on page 51).

If you accept the default, the aliases are created and the installation program displays the following prompt:

```
Alias /iw-mount/ /iwmnt/
Alias /iw-mount/ has been added to /usr/local/apache/conf/srm.conf
Restart your Web Server to be ready for use? [y] y
```

15. Press **Return** to accept the default and restart your Web server.

```
Web server restarted successfully
```

TeamSite can do this automatically if your http server has a restart script and you type **y** when prompted. You can bypass this step by typing **n** and manually restart the Web server later.

The installation program then displays the option to install Samba or PCnfsd.

```
STEP 10: Install Samba or PCnfs authenticator [OPTIONAL]
If you plan to connect to Interwoven TeamSite from a PC running Windows,
and you do not have Samba or PCnfs installed, you will need to install
one of them now. We recommend that you install Samba because it does not
require client-side software.
```

```
Do you want to install Samba now? [y] y
```

16. Press **Return** to accept the default and install Samba, or type **n** to display the option to install PCnfsd (answer **n** to this prompt if you want to continue without installing either option).

If you install Samba, you are prompted for a workgroup name. The remainder of the Samba installation is automatic. If you decide to install PCnfsd, the entire installation is automatic, but requires client-side software.

```
Reconfiguring Samba Server TeamSite mount points...      Done
Which Windows workgroup do you want the Interwoven TeamSite server to
appear in?
Please enter WORKGROUP name: [interwoven]
Not installing TeamSite-supplied PCNFSD: /etc/rc3.d/S50iw.pc nfsd
```

17. Press **Return** to accept the default workgroup name or enter another name and press **Return**.

You are prompted to install the Global Report Center which generates reports that show common user activities in any TeamSite workarea or branch:

```
STEP 11: Install TeamSite Global Report Center
Do you want to install the TeamSite Global Report Center? [y] y
Installation of TeamSite Global Report Center was successful.

Created /iw-home/log/deployEvents.log
Preparing the Global Report Center. Please wait...
Creating the jdbc stored procedures...
```

18. Press **Return** or type **y** at the prompt to install the reporting functionality.

The TeamSite installation program then automatically installs the OpenAPI service and begins the Web daemon installation by prompting you to specify a series of port numbers.

Note: The HTTP server is assigned port 80 by default. In past releases of TeamSite, port 80 was assigned to the Web server by default. If you do not use port 80 for the HTTP server, users must explicitly specify the alternate port number in the URL each time they access TeamSite. Also note that the default port number for the Web server is 81.

STEP 12: Setup OpenAPI Service

OpenAPI Service has been set up successfully.

STEP 13: Setup Interwoven Web Daemon

Enter HTTP port number that the web browser will use to contact the TeamSite server when using the TeamSite web interface if it is different from the default value. [80]

Enter HTTPS port number that the web browser will use to contact the TeamSite server when using the TeamSite web interface if it is different from the default value. [443]

Enter the fully qualified host name that the web browser will use to contact the TeamSite server when using the TeamSite web interface if it is different from the default value. [colts.interwoven.com]

Enter the port number for application server servlet engine if it is different from the default value. [8080]

Enter the port number for Interwoven proxy server if it is different from the default value. [1080]

Enter your web server port number if it is different from the default value. [81]

19. To accept the default values for each port number, press **Return**.

The Web Daemon configuration is confirmed:

Interwoven Web Daemon has been configured.

To complete the installation procedure, TeamSite looks for a valid license key.

- If you have a valid license key, apply it as described in “Installing the License Key” on page 47.
- If you do not have a valid license key, obtain one as described in “Obtaining a License Key” on page 47.

Installing TeamSite 5.5.2 on AIX

This section describes the TeamSite installation process. Ensure that you have satisfied the hardware and software requirements described in the preceding sections before starting the installation program. The installation process is recorded in the `iwinstall.log` file in `iw-home/install/`.

If you are upgrading to TeamSite 5.5.2, proceed to “Upgrading to TeamSite 5.5.2” on page 49.

To install TeamSite on your Web site development server:

1. Log in as `root` to the system where you want to install the TeamSite server.

Note: Do not use `sudo su -` or `su` to gain root access when installing TeamSite. Using `sudo su -` or `su` instead of `su -` may cause problems with shutdown, startup, or installation operations.

2. Insert the TeamSite CD-ROM and browse to the top level directory.
3. Copy the installation file, `IWOVts-aix5.5.2.BuildNumber.tar.gz`, to the directory in which you want the TeamSite application to reside.
4. Before running the `tar` command to uncompress and expand the TeamSite installation for AIX, run the following command:

```
ulimit -f unlimited
```

5. Uncompress and expand the `IWOVts-aix5.5.2.BuildNumber.tar.gz` file by running the following command:

```
% gunzip -c IWOV-aix5.5.2.BuildNumber.tar.gz |(cd /parent_directory;  
tar xvzf -)
```

The default `parent_directory` for TeamSite is `/usr`. This command creates a directory called `iw-home` under the `parent_directory`. The `iw-home` directory contains the TeamSite program files.

6. Locate the TeamSite installation file `iwinstall` in the subdirectory `iw-home/install`. From this directory, run the following command to starts the TeamSite installation process:

```
% ./iwinstall
```

The following message and prompt is displayed:

```
Starting Installation of Interwoven TeamSite (R)
iwininstall version <5.5.2 Build 10013 SYM Interwoven 20020717>
Start Time: Fri Jul 19 14:45:16 PDT 2002

Copyright 1997-2002 Interwoven, Inc. All rights Reserved.

ADA Software Use Restriction: This Sybase Adaptive Server Anywhere
("ADA") software module embedded in TeamSite is licensed from Sybase,
formerly called "SQL Anywhere". The Sybase application software
license strictly limits the use of ADA software by end users to
Global Report Center. End User is prohibited from using this ADA
software module on any application portion of TeamSite, outside of
Global Report Center. Any violation of this use restriction will
subject end user to a termination of this ADA license.
```

Do you agree with the license agreement (y/n)? [n] y

7. Type **y** and then press **Return** to accept the license agreement.

The installation program checks to ensure there is sufficient disk space for TeamSite:

```
TeamSite Installation needs at least 300MB of free disk space.
Current disk space on this machine is:
Filesystem      1024-blocks      Free  %Used      Iused %Iused Mounted on
/dev/hd4          1048576    817336   23%       3762   1%   /
/dev/hd2          5242880    225120   96%      97536   8%   /usr
/dev/hd9var        262144    209416   21%       939   2%   /var
/dev/hd3          2097152    2031196   4%        35   1%   /tmp
/dev/hd10opt       2097152    1594252   24%      15117   3%   /opt
/dev/lv00          20971520   15912920  25%      26857   1%   /data
Do you wish to continue (y/n)? [y]
```

8. Press **Return** to accept the default, or type **y**.

The following message and prompt is displayed:

```
Creating Interwoven TeamSite UI Daemons User (iwui)...
Please choose a UID for the new user, or
press <Enter> to accept the next available UID: [<Enter>]
```

9. Press **Return** to accept the default.

The installation program confirms the creation of the `iwui` user, copies the TeamSite program files and creates the TeamSite directory:

This may take a moment ...

The user "iwui" has been created successfully.

STEP 0: Copying platform specific files
Creating directory /etc/rc.d/init.d
Installing /etc/rc.shutdown
Interactively installing...

TeamSite server and related processes are stopped

STEP 1: Install default IW_HOME directory.
Creating new /etc/defaultiwhome ...
IW_HOME is now /usr/iw-home
Installing /usr/bin/iwgethome
Moving obsolete file/directory to /usr/iw-home/OBSOLETE/iwdeploy
Configuring TeamSite executables. This will take a few minutes.
Configuring bin/XalanTransform... done.
Configuring bin/iwconvert... done.
Configuring bin/iwcpfile... done.
Configuring bin/iwcpwa... done.
Configuring bin/iwgetldapusers... done.
Configuring bin/iwmigrate... done.
Configuring bin/iwproxy... done.
Configuring bin/iwserver.aix5... done.
Configuring httpd/iw-bin/iwmerge.cgi... done.
Configuring iwopenapi/libopenapijni.so... done.

STEP 2: Not applicable to AIX

STEP 3: Check kernel extension

STEP 4: Install and export /iwserver
/iwserver has been added to the NFS exports

STEP 5: Install software in /etc ...
Installing items in /etc/rc.d/init.d...
Installing iwserver to start at boot time: /etc/rc.d/rc2.d/S16iw.server
Installing iwserver to stop at shutdown time: /etc/rc.d/rc6.d/K16iw.server
Installing iw.local to start at boot time: /etc/rc.d/rc2.d/S99iw.local
Installing iw.local to stop at shutdown time: /etc/rc.d/rc6.d/

```
K99iw.local  
Installing iw.reboot to start at boot time: /etc/rc.d/rc2.d/  
S15iw.reboot
```

You are prompted to specify locations for the TeamSite log files, mount point, and backing store.

```
STEP 6: Install default configurations in /etc/defaultiw... files  
Enter the location to store log files [/var/adm]:  
iwevents.log, iwtrace.log, iwserver.log and other logs are located in  
/var/adm  
Accept default mount directory location /iwmnt (y/n)? [y] y  
Default entry /iwmnt saved in file /etc/defaultiwmount  
Accept default backing store directory location /local/iw-store  
(y/n)? [y] y  
Default entry /local/iw-store saved in file /etc/defaultiwstore
```

10. Press **Return** to accept each of the default locations, or type **n** at the prompt to specify a different location.

Note: If you specify an alternate location for the backing store, you must use ASCII characters.

The system confirms the creation of the log files and the backing store and writes that location of the backing store to the `/etc/defaultiwstore` file:

```
Default entry /local/iw-store saved in file /etc/defaultiwstore
```

This file tells TeamSite where the backing store is located.

The installation program continues by installing the Perl programs:

STEP 7: Install Perl programs.
Installing /usr/iw-home/iw-perl
It will take a few minutes...
Creating /usr/iw-home/local/config/file_encoding.cfg
Creating /usr/iw-home/local/config/wft/available_templates.ipl
Creating /usr/iw-home/local/config/wft/available_templates.cfg
Creating /usr/iw-home/events subsystem/conf/jmsconfig.xml
Creating /usr/iw-home/events subsystem/conf/events subsystem.properties
Creating /usr/iw-home/local/config/wft/solutions/
configurable_default_submit.cfg
Creating /usr/iw-home/local/config/wft/solutions/
configurable_author_submit.cfg
Creating /usr/iw-home/local/config/wft/solutions/
configurable_author_assignment.cfg

The installation program continues to create and populate access control groups:

STEP 8: Create group members for access control
The following files will contain group membership information for
Administrators, Editors and Authors. YOU MAY MODIFY THEM BEFORE OR
WHILE Interwoven TeamSite IS RUNNING BY EDITING THE FILES DIRECTLY.

/usr/iw-home/conf/roles/master.uid
/usr/iw-home/conf/roles/admin.uid
/usr/iw-home/conf/roles/editor.uid
/usr/iw-home/conf/roles/author.uid
/usr/iw-home/conf/roles/od-admin.uid
/usr/iw-home/conf/roles/od-user.uid
/usr/iw-home/conf/roles/content-provider.uid
/usr/iw-home/conf/roles/syndication-admin.uid

Please select your option for initial <master> group setup
[1] Assign everyone in my /etc/passwd file to be an master.
[2] I'd like to manually assign a few members now.

Please enter your choice: [1]

11. Press **Return** to accept the default and automatically assign all members listed in the password file to the specified TeamSite role (master), or type **2** and manually specify individual users for each role.

When manually assigning user names for a role, type each name separated by a space, then press **Return** when you are finished.

You can modify these files at any time (see page 84). Deleting a user name from a file removes TeamSite access privileges for the role. Adding a user name to a file grants access privileges for the role.

The system adds the specified users to the `master.uid` role file and prompts you to specify users for the `admin.uid`.

```
Installing master roles in /usr/iw-home/conf/roles
Done creating roles in /usr/iw-home/conf/roles
Please select your option for initial <admin> group setup
[1] Assign everyone in my /etc/passwd file to be an admin.
[2] I'd like to manually assign a few members now.

Please enter your choice: [1]
```

12. Repeat step 11 for each of the remaining role files.

After the final group is created, the following prompt is displayed:

```
STEP 9: Install URL alias for Apache Content Web Server
Are you using Apache as your Content Web Server (y/n)? [y]
```

13. If you are using an Apache Content Web Server, press **Return** or type **y** to install a series of aliases that enable you to access TeamSite directly from within the Web server.

If you do not want TeamSite to install these aliases, type **n**. If you are using a Netscape Web server, type **n** (continue with step 15).

You are then prompted to confirm the `httpd` executable and configuration directories.

```
Is "/opt/freeware/apache/sbin" your httpd executable directory
(y/n)? [y]
Is "/etc/opt/freeware/apache" your httpd configuration directory
(y/n)? [y]
```

14. Press **Return** to accept the defaults, or type **n** and enter the Web server `httpd` locations (if you type **n**, you will need to configure your Web server manually after the TeamSite installation is finished as described on page 51).

If you accept the defaults, the aliases are created and the installation program displays the following prompt:

```
Alias /iw-mount/ /iwmnt/
Alias /iw-mount/ has been added to /etc/opt/freeware/apache/
httpd.conf

Restart your Content Web Server to be ready for use (y/n)? [y]
```

15. Press **Return** to accept the default and restart your Web server.

```
/opt/freeware/apache/sbin/apachectl restart: httpd restarted
Content Web server restarted successfully
```

TeamSite can do this automatically if your http server has a restart script and you type **y** when prompted. You can bypass this step by typing **n** and manually restart the Web server later.

The installation program then displays the option to install Samba.

```
STEP 10: Install Samba [OPTIONAL]
If you plan to connect to Interwoven TeamSite from a PC running
Windows, and you do not have Samba installed, you will need to
install it now.
Do you want to install Samba Version 2.2.0a now (y/n)? [y] y
```

16. Press **Return** to accept the default and install Samba, or type **n** to continue without installing Samba.

If you install Samba, you are prompted for a workgroup name. The remainder of the Samba installation is automatic.

```
Reconfiguring Samba Server TeamSite mount points...      Done
Which Windows workgroup do you want
the Interwoven TeamSite server to appear in?
Please enter WORKGROUP name: [interwoven]
```

17. Press **Return** to accept the default workgroup name or enter another name and press **Return**.

You are prompted to install the Global Report Center which generates reports that show common user activities in any TeamSite workarea or branch:

```
STEP 11: Install TeamSite Global Report Center
Do you want to install the TeamSite Global Report Center (y/n)? [y] y
Created database directory
Creating system tables
Collation sequence: ISO_1
Creating system views
Setting permissions on system tables and views
Setting option values
Loading Java classes
Initializing UltraLite deployment option
Database "/data/iw-home/report/db/server.db" created successfully
Created new log database.
Installation of TeamSite Global Report Center was successful.
Created /data/iw-home/log/deployEvents.log
Preparing the Global Report Center. Please wait...
Creating the jdbc stored procedures...
```

18. Press **Return** or type **y** at the prompt to install the reporting functionality.

The TeamSite installation program then automatically installs the OpenAPI service and begins the Web daemon installation by prompting you to specify a series of port numbers.

Note: The HTTP server is assigned port 80 by default. In past releases of TeamSite, port 80 was assigned to the Web server by default. If you do not use port 80 for the HTTP server, users must explicitly specify the alternate port number in the URL each time they access TeamSite. Also note that the default port number for the Web server is 81.

STEP 12: Setup OpenAPI Service

OpenAPI Service has been set up successfully.

STEP 13: Setup Interwoven Web Daemon

Enter HTTP port number that the web browser will use to contact the TeamSite server when using the TeamSite web interface: [80]

Enter HTTPS port number that the web browser will use to contact the TeamSite server when using the TeamSite web interface: [443]

Enter the fully qualified host name that the web browser will use to contact the TeamSite server when using the TeamSite web interface: [factorum.example.com]

Enter the port number for Interwoven Servlet Engine: [8080]

Enter the port number for Interwoven proxy server: [1080]

Enter your Content Web Server port number: [81]

19. To accept the default values for each port number, press **Return**.

The Web Daemon configuration is confirmed:

Ports used are:

HTTP Port (80)

HTTPS Port (443)

Interwoven Servlet Engine Port (8080)

Interwoven Proxy Server Port (1080)

Content Web Server Port (81).

Interwoven Web Daemon has been configured.

Installing [tsadmin.war] as webapp tsadmin... Configuring JSPs... . . . done.

Executing /usr/iw-home/httpd/webapps/installinfo/tsadmin/WEB-INF/post-install_webapp.ipl

Installing [eventsubsystem.war] as webapp eventsubsystem... Configuring JSPs... . . . done.

Installing [dctwidgetlib.war webdesk.war] as webapp webdesk... Configuring JSPs... . . . Note: Some input files use or override a deprecated API.

Note: Recompile with -deprecation for details.

. done.

Webapps successfully expanded.

To complete the installation procedure, TeamSite looks for a valid license key.

- If you have a valid license key, apply it as described in “Installing the License Key” on page 47.
- If you do not have a valid license key, obtain one as described in “Obtaining a License Key” on page 47.

Obtaining a License Key

Complete the following procedure to obtain a license key when prompted by the TeamSite installation program:

1. Enter an email address where you can receive a license key, then press **Return**.
2. Open the `tslicinfo.log` file in `iw-home/install` to obtain the information you will need to generate your license key.

```
hostname factotum
platform UX
domain example.com
hostMACid 80daac37
email admin@example.com
product TS
version 5.5.2
os_version SunOS 5.8 [optional]
```

For AIX, `os_version` is AIX 5.1 [optional].

3. Log on to the Interwoven Support Web site's license generator page at:
<https://support.interwoven.com/license/license.asp>
Follow the directions for obtaining a license key. After you enter the required fields, Interwoven will send a license key to the specified email address.

Installing the License Key

After you receive your license key from Interwoven support, you will need to install it. You can do this two ways:

- Entering it when prompted by the installation program.
- Entering it in the `iw.cfg` file.

Regardless of which method you use, the license information is stored in the `[iwserver]` section of the `iw.cfg` file using the following format:

```
license_expires=date (yyymmdd format)
license_key=key * host * version * email_address
```

To install the license key when prompted by the installation program:

1. Enter the license expiration date at the prompt, for example:

```
license_expires=20010701
```

2. Enter the license key at the prompt, for example:

```
license_key=bb772957a7b7c491a841aa958c5a3 * host * 5.5.2 * admin@example.com
```

3. Reboot your system.

To enter your license key in the `iw.cfg` file:

1. Locate your `iw.cfg` file using the `iwgetlocation` command-line tool (located in `iw-home/bin`), for example:

```
% iwgetlocation -c iwconfig
```

The command in the example would return the following (default) location:
`/etc/iw.cfg`

2. Using a text editor, add the following lines to the `[iwserver]` section of the `iw.cfg` file as shown in the following example (or copy and paste the `license_expires` and `license_key` lines from the license key email):

```
[iwserver]
license_expires=date
license_key=key * host * version * email_address
```

Some types of licenses may require additional lines to be added to the `iw.cfg` file. You will receive further information with your license key, if necessary.

3. Save and close the file.
4. Reboot your system.

Troubleshooting Licensing Issues

If your TeamSite server fails to run after you have installed your license key:

- Ensure the entry in the [iwserver] section of your `iw.cfg` file matches the email sent from Interwoven.
- Use the `tsisvalid` command-line utility (located in `iw-home/bin`) to verify that the license installed in `iw.cfg` is valid:

```
% tsisvalid iw-home
```

The `tsisvalid` command creates a license status report file in `iw-home/install/tsisvalid.log`. If the license in the `iw.cfg` file is valid, `tsisvalid` prints the following line in the report file:

`License is good.`

If your license key is invalid, `tsisvalid` will print a report of possible reasons why it was not able to validate the license.

- Look for diagnostic messages in the `iwserver.log` and `iwtrace.log` files (located by default in `iw-home/local/logs`).

Upgrading to TeamSite 5.5.2

The TeamSite 5.5.2 upgrade procedure is the same as it has been for previous releases, but the circumstances under which you upgrade are dependant on the backing store conversion. In past releases you would load the new release onto the system where your current TeamSite server was installed. For TeamSite 5.5.2, the upgrade scenario is as follows:

1. With TeamSite 4.5.x or TeamSite 5.x installed on your current system (call it system A), install TeamSite 5.5.2 on a second system (system B). See “Installing TeamSite 5.5.2 on Solaris” on page 30 or “Installing TeamSite 5.5.2 on AIX” on page 38 for instructions.
2. Convert your old-format backing store (on system A) to the new high-performance backing store format on the system B. (This procedure is described in Chapter 8, “TeamSite Backing Stores”.)
3. If you are upgrading TeamSite on AIX, and you are using the installation of Samba that came with TeamSite, stop the Samba server:
`# /etc/init.d/iw.samba stop`

4. If you want to use System A as your TeamSite 5.5.2 deployment server (as you have been), you must upgrade System A to TeamSite 5.5.2 and also migrate the new backing store (or stores) to system A.

Note: When copying the new backing store from one machine to another, you must ensure that all file attributes—including security and file times—are preserved.

The procedure for upgrading is described later in this section.

Note the following:

- The process of extracting TeamSite's installation files will overwrite all files contained in `iw-home` and its subdirectories, except for files that are not part of the default TeamSite installation.
- The contents of the `iw.cfg` file have changed for the 5.5.2 release. During the upgrade process, a copy of your existing `iw.cfg` file will be automatically renamed `iw.cfg.4.5.x` or `iw.cfg.5.0.x` and new `iw.cfg` and `iw.cfg.example` files will be created. If you had any customized settings in the previous version of your `iw.cfg` file that you want to apply to TeamSite 5.5.2, you will need to manually enter them into the new `iw.cfg` file.
- The default HTTP port numbers may have changed since you installed TeamSite.
- If you are upgrading TeamSite Templating, the `available_templates.ipl` file is no longer used. It has been replaced by a new file called `available_templates.cfg`. If you modified the `available_templates.ipl` file and want to apply them to the 5.5.2 release, you will need to manually add your changes to the new `available_templates.cfg` file.
- If you have an existing TeamSite installation earlier than version 4.5, you will need to install the appropriate TeamSite patch. For information on obtaining TeamSite patches, contact your Interwoven sales representative, or <http://support.interwoven.com>

To upgrade to TeamSite 5.5.2:

1. Back up your TeamSite `iw-store` directory, and your existing TeamSite configuration files, roles files, and any customized site-specific files. These files include the following:
 - `iw-home/etc/iw.cfg`
 - `iw-home/local/config/templates.cfg`
 - `iw-home/local/config/autoprivate.cfg`
 - `iw-home/conf/roles/*.*uid`

- Any server start or stop scripts in the `/etc/init.d` directory
 - Any other customized files such as custom login screens or a custom-configured `iw-home/private/bin/iwetcboot` file
2. Log on to your system as `root`.
 3. Stop the TeamSite server with one of the following commands:
 - If upgrading a Solaris system from TeamSite 5.5.1 or earlier release, use:
`% /etc/init.d/iw.server stop`
 - In all other cases, use:
`% /etc/init.d/iw.server stop_all`
 4. Stop your Web server.
- Note:** Do not use `sudo su -` to gain root access when installing TeamSite. Using `sudo su -` instead of `su -` may cause problems with shutdown, startup, or installation operations.
5. Continue the installation procedure as described in “Installing TeamSite 5.5.2 on Solaris” on page 30 or “Installing TeamSite 5.5.2 on AIX” on page 38.
- When the installation script asks you whether you want to upgrade TeamSite, type **y**.
If you are upgrading to TeamSite 5.5.2L on AIX, reboot the system after installing TeamSite and before starting the TeamSite server.
- Several Web server configuration changes have been made to the TeamSite 5.5.2 release. After the upgrade process is completed, proceed to the next section, “Configuring Web Servers.”

Configuring Web Servers

This section describes how to configure your system’s Web server after you have installed or upgraded TeamSite. Configuration procedures for the following Web servers are included:

- Apache
- Netscape Enterprise Server
- iPlanet

Note: You must stop and restart your Web server to activate the changes described in the following sections.

Specifying the Web Server Port Number

The Web server port number specified in your Web server's `httpd.conf` file must match the port number assigned in the `iw.cfg` file during the TeamSite installation. By default, TeamSite assigns port 81 to the Web server. If you are using the default configuration, ensure that the port number in your `iw-home/iw-webd/conf/httpd.conf` file reads as follows:

```
# Port: The port to which the standalone server listens. For ports
# <1023, you will need httpd to be run as root initially.
#
Port 81
```

Specifying the Web Server HTTPD User Name

When TeamSite is installed, the user name `nobody` is added to the `master.uid` file in `iw-home/conf/roles/`. If the HTTPD user name for your system is not `nobody`, you must specify the correct user name in the `master.uid` file.

For information about how to edit the `master.uid` file, see “Adding and Removing Users” on page 84.

Configuring the `iw-mount` Alias

The `iw-mount` alias enables your Web server to access the NFS server mount point. If you are upgrading to TeamSite 5.5.2, the `iw-mount` alias now incorporates the functionality of the former `iw`, `iw-bin`, and `iw-icons` aliases. The process for configuring `iw-mount` varies depending on the type of Web server you are using.

Configuring `iw-mount` for NES or iPlanet

Configure the `iw-mount` alias for NES or iPlanet using the Server Administrator.

To configure `iw-mount` using the NES or iPlanet Server Administrator:

1. Open the Server Administrator.

The Server Administrator is displayed.

2. In the **Manage Servers** section of the Server Administrator, select the appropriate server name and click **Manage**.

The Server Administrator returns the status of the selected server.



3. At the top of the Server Administrator, click the **Content Mgmt** tab.

The Content Management screen is displayed.

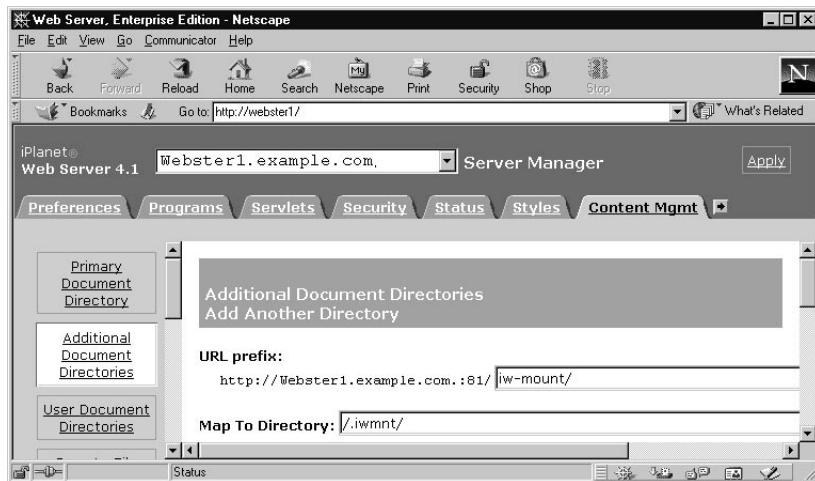
4. On the left panel of the Server Administrator, click **Additional Document Directories**.



The Additional Document Directories screen is displayed.

5. In the Additional Document Directories screen, make the following entries:

- In the **URL Prefix** field, enter **iw-mount/**
- In the **Map To Directory** field, enter **/ .iwmnt/**



6. Click **OK**, and save your changes when prompted.

Configuring iw-mount for Apache

To configure the **iw-mount** alias for Apache, enter the following section in the **httpd.conf** file:

```
<Directory "/.iwmnt">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
Alias /iw-mount/ /.iwmnt/
```

Configuring CGI Programs

If your system will be executing CGI programs in TeamSite workareas, you will need to edit the `httpd.conf` file. After making the required edits, you must restart the Web server for the changes to take effect. If you reinstall your Web server or lose your automatically modified configuration files, you can refer to the installation log to review the changes previously made.

For NES and iPlanet, add the following section to the `obj.conf` file:

```
<Object ppath="/.iwmnt/*/cgi-bin/*">
  ObjectType fn="force-type" type="magnus-internal/cgi"
  Service fn="send-cgi"
</Object>
```

For Apache, perform the following steps:

1. Add the `ExecCGI` directive to the `Directory` section in the `httpd.conf` file as shown below:

```
<Directory "/.iwmnt">
  Options Indexes FollowSymLinks ExecCGI
  AllowOverride None
  Order allow,deny
  Allow from all
</Directory>
```

2. Add the following section:

```
<Location ~ "/iw-mount/.*/cgi-bin/.*"/>
  SetHandler cgi-script
</Location>
```

Enabling Server-Side Include Requests

Because server-side include requests (SSIs) do not go through the proxy server, you must install the TeamSite redirector module to enable SmartContext QA for SSIs. The configuration steps vary depending on the type of Web server you are using.

You must configure your web server to use server-side includes by specifying that it parses .shtml files. For more information on this process, consult the NCSA server-side include tutorial at:

<http://hoohoo.ncsa.uiuc.edu/docs/tutorials/includes.html>.

If your Web site does not use SSIs, you do not need to install the redirector module.

Installing the Redirector Module for NES and iPlanet

1. Open the `obj.conf` file in your text editor.

The file is located in `iw-home/servletd/conf/`.

2. In the `Init` section, add the two `Init` directives shown below, substituting the pathname to the `iwrewrite.nsapi.solaris.so` file as appropriate for your installation.

First entry (all on one line):

```
Init fn="load-modules" shlib="/installation_directory/iw-home/lib/  
iwrewrite.nsapi.solaris.so" funcs="iwrewrite"
```

Second entry :

```
Init fn="iwrewrite"
```

- a. In the `default` object description, add the following `NameTrans` as the first `NameTrans`, superseding all others:

```
NameTrans fn="iwrewrite"
```

Installing the Redirector Module for Apache

If your system uses an Apache Web server, there are two different types of redirector modules available. The type of module you use will depend on your configuration and is determined by the procedure that follows. Complete the procedure to install and configure the appropriate redirector module for Apache:

1. Determine if `mod_so` is enabled by running the following directive:

```
#apache-home/bin/httpd -l
```

2. Look for `mod_so.c` in the output, then do one of the following:

- If `mod_so.c` is in the output, proceed to Step 3.

- If mod_so.c is not in the output, run the following configuration directive to configure and build Apache with mod_so:

```
# ./configure --enable-shared=max --prefix=/path_to_apache_home  
# make  
# make install
```

3. On AIX, your Apache web server must be linked using the -brtl link option to enable run-time linking. To determine if Apache is linked with the -brtl option, use the command:

```
$ dump -H httpd | grep librt
```

If you see a line in the output containing librt.a, then Apache is linked with the -brtl option. If not, re-link Apache to include the option.

4. Determine if your Apache Web server has a mod_ssl patch by running the following command:

```
httpd -V
```

5. Look for -D EAPI in the output.

- If the output contains -D EAPI, use the mod_iw_ts_rewrite_eapi.so module.
- If the output does not contain -D EAPI, use the mod_iw_ts_rewrite.so module.

6. Edit the LoadModule section of the httpd.conf file as follows:

- If you are using the mod_iw_ts_rewrite_eapi.so redirector module:

```
LoadModule iw_ts_rewrite_module iw-home/lib/mod_iw_ts_rewrite_eapi.so
```

- If you are using the mod_iw_ts_rewrite.so redirector module:

```
LoadModule iw_ts_rewrite_module iw-home/lib/mod_iw_ts_rewrite.so
```

7. Enable relative SSIs in .shtml files by doing the following:

- a. Add the Includes statement to the Directory section of the httpd.conf file as shown in the following example:

```
<Directory "/.iwmnt">  
    Options Indexes FollowSymLinks ExecCGI Includes  
    AllowOverride None  
    Order allow,deny  
    Allow from all  
</Directory>
```

- b. Add or uncomment the <If Module mod_mime.c> section in the httpd.conf file as shown in the following example:

```
<If Module mod_mime.c>
AddType text/html .shtml
AddHandler server-parsed .shtml
</IfModule>
```

8. Edit the end of the AddModule section of the httpd.conf file as shown in the following example:

```
AddModule mod_iw_ts_rewrite.c
```

Stopping and Restarting the Web Server

Before you activate your configuration changes by rebooting your Web server, ensure that the docroot lines in the [iwproxy_remap] section of the iw.cfg file do not end with a trailing slash. For example:

```
[iwproxy_remap]
extranet_branch=/main/extranet
global_default_map=/

[extranet_branch]
_docroot=/htdocs ← No trailing slash allowed
/icons=/icons/ ← Trailing slashes allowed
/images=/multimedia/img/ ←
```

After ensuring that all _docroot trailing slashes in the [iwproxy_remap] section are deleted, stop and restart your Web server.

Redirecting NSAPI HTTPS Requests

You can configure TeamSite to redirect HTTPS requests from TeamSite so they are served from the Web daemon (iwwebd) over HTTP. To do this, your system must contain two Web servers:

- A secure NES or iPlanet Web server set up to process HTTPS requests.
- A non-secure server of any type that processes TeamSite web daemon HTTP requests.

To redirect HTTPS requests, set the following directive in the [nsapi] section of your `iw.cfg` file:

```
redirect_https_to_http=yes
```

When redirection is enabled, all HTTPS requests originating from the browser and received by the secure server's NSAPI plugin are redirected to the web daemon. The web daemon then sends the requests to the non-secure TeamSite server just as it would any request originating from the browser. For example, if the NSAPI plugin on the secure server receives an HTTPS request for a file in a TeamSite area such as:

```
https://teamsite_host/iw-mount/branch1/STAGING/bio.html
```

where `teamsite_host` specifies the host name (such as `www.example.com`), then the request is redirected to the web daemon as follows:

```
http://teamsite_host:iwwebd_port/iw-mount/branch1/STAGING/bio.html
```

where `iwwebd_port` specifies the port number.

During the redirection process, some browsers will display a message warning that the request is being sent to an insecure document. This is normal browser behavior. If you see such a message, click **OK** to proceed. Note that HTTPS requests redirected to the web daemon no longer have HTTPS security.

Configuring Samba

TeamSite includes an optional installation for Samba, a network protocol that makes it possible for PCs to mount UNIX drives as Windows networked drives. Samba is a server-side solution which does not require client software. This makes TeamSite easy to administer because the system administrator does not need to update and correct issues with client computers.

The default TeamSite Samba configuration file is `iw-home/samba/lib/iw.smb.conf`. If you already have Samba installed, the file could be in `/usr/local/samba/etc/smb.conf` or `/usr/local/samba/lib/smb.conf`.

Edit your Samba configuration file as follows:

1. Set the following global option to grant access privileges on a user-by-user basis:

```
security = user
```

2. Add the following section to your Samba configuration file:

```
[iwmain]
comment = directory main branch of TeamSite file system
public = no
create mode = 0775
force create mode = 0775
force directory mode = 0775
writable = yes
locking = no
share modes = no
preserve case = yes
short preserve case = yes
path = /iwmnt/default/main
```

Note the following:

Directive	Description
<code>public=no</code>	Specifies that authentication is required to view the contents of the Samba mount.
<code>create mode=0775</code>	Sets UNIX permissions on all newly created files to 0775.
<code>force create mode=0775</code>	Sets the minimum UNIX permissions on all newly created files to 0775.
<code>force directory mode=0775</code>	Sets the minimum UNIX permissions on all newly created directories to 0775.
<code>writable=yes</code>	Allows users to write to the Samba mount (<code>writable=no</code> disables the ability to do writes via Samba).
<code>locking=no</code>	Turns off locking. Locking must be turned off, because TeamSite has its own locking.
<code>share modes=no</code>	Turns off SMB deny modes.
<code>preserve case=yes</code>	Keeps the original case of file names.
<code>short preserve case=yes</code>	Keeps the case of 8.3 filenames rather than displaying them as all uppercase.
<code>path=/iwmnt/default/main</code>	The directory to be mounted.

3. Save and close the `iw.smb.conf` file.

4. Restart Samba for your changes to take effect as shown in the following example:

```
% /etc/init.d/iw.samba stop  
% /etc/init.d/iw.samba start
```

To mount any other TeamSite area:

1. Copy and paste the [iwmain] section in your Samba configuration file.
2. Change the name of the section from [iwmain] to a unique identifier.
3. Edit the path line to point to the area to be mounted.
4. Restart Samba for your changes to take effect.

Troubleshooting Samba

If Samba is not running, ensure that the version of Samba that is installed with TeamSite provides a link between `/usr/local/samba/lib/smb.conf` and `iw-home/samba/lib/iw.smb.conf`.

If the `/usr/local/samba/lib` directory does not exist, complete the following procedure:

1. Create the directory:
`% mkdir -p /usr/local/samba/lib`
2. Create the `iw.smb.conf` file if it does not already exist:
`% cp `iwgethome`/samba/lib/iw.smb.conf.ex `iwgethome`/samba/lib/
iw.smb.conf`
3. Link the file:
`% ln -s `iwgethome`/samba/lib/iw.smb.conf /usr/local/samba/lib/
smb.conf`
4. Test the parameters:
`% `iwgethome`/samba/bin/testparm`
5. Start `iw.samba`:
`% /etc/init.d/iw.samba start`

Configuring TeamSite Clients

After installing TeamSite and configuring your Web server, you will need to set up at least one TeamSite client. You can use either the graphical user interface or the file system interface for client access.

Using the Graphical User Interface

TeamSite's graphical user interface can be accessed through a JavaScript-capable web browser. In order to log in, you must be a TeamSite user. If you have not yet added users to TeamSite or changed your own user status, you should do so now (see Chapter 3, "Managing Access"). If you do not add users or change your own user status, you are limited to the TeamSite Master role, which is the default role for the user `root`.

For detailed information about browser configuration, consult the *TeamSite User's Guide*.

1. To access TeamSite from a client computer, start your web browser and enter the following URL:

`http://TeamSite_hostname/iw/`

where `TeamSite_hostname` is the name of the TeamSite server (for example, `teamsite1.example.com`). You may want to bookmark this URL for future use.

The TeamSite login screen is displayed.



2. Select your user type (Author, Editor, Administrator, or Master) from the Login as menu.
3. If you are logging in as an Author, select the desired interface:
 - WebDesk—Updated TeamSite interface that provides superior usability.
 - WebDesk Pro—Provided for users who are familiar with earlier versions of TeamSite.

To use WebDesk, ensure you selected the Author role in step 2, and check the **WebDesk** check box. To use WebDesk Pro, proceed to step 4.
4. Enter your UNIX user name and password and click **Login** to display the selected user interface.

Installing LaunchPad

Before you can edit files using TeamSite, you will need to install LaunchPad (the TeamSite helper application). If you are already using an older version of LaunchPad, you will be prompted to upgrade). To install LaunchPad:

1. Log in to TeamSite through the browser interface.
2. Select **Edit > LaunchPad Setup**.

See the “Getting Started” chapter of the *TeamSite User’s Guide* for more information on installing, configuring, and using LaunchPad.

Using the File System Interface

The file system interface enables you to manage your web content in TeamSite as if it were on a mounted drive on the network. The file system interface is used primarily for file management functions such as moving and copying files, and it can also be used to edit files. It also allows the use of link checkers and scripts that need to be able to access or create files. In addition, most TeamSite operations can be performed from a UNIX command-line interface (see the documentation on Command Line Tools).

Windows 98, 2000, and NT Clients

The first time you access your TeamSite server from a Windows client, you may need to mount the TeamSite server as a network drive. The following procedure describes how to access TeamSite with:

- A networked computer via Samba or a NFS client.

- A networked computer with FTP.

To access TeamSite from Windows via Samba, use Windows Explorer to locate the TeamSite server.

1. Click **Start > Search > For Files or Folders**.
2. Click the **Search for other items: Computers** option.
3. Type the name of the TeamSite server in the **Computer Name** field.
4. Click **Search Now**.
5. Double-click the name of the TeamSite server when it appears in the list.
The server window is displayed.
6. Double-click the TeamSite mount point directory (usually `iwmain`) and navigate through the TeamSite directory structure to find your workarea.
Within your workarea, you can edit, move, or rename TeamSite files as you would any other files. You can also drag and drop files and directories from your local hard drive to directories in your workarea.

Creating Shortcuts and Mapping Network Drives

To simplify future access to your workarea or to commonly used directories in your workarea, create a shortcut to the directory or directories you access frequently and put it on your desktop.

To simplify future access to the TeamSite server, and to enable LaunchPad Direct Edit (see the *TeamSite User's Guide*), map a network drive to the TeamSite server.

Troubleshooting Windows Networking

If you cannot find the TeamSite server through Windows Networking, check to see if you have NetBEUI installed. If you do, uninstall it if at all possible.

To uninstall NetBEUI:

1. Select **Start > Settings > Control Panels**.
2. Double-click the **Network** control panel icon.

3. Select the **Protocols** tab.
4. From the list of adapters and their associated protocols, find the local client Ethernet card adapter. If NetBEUI precedes TCP/IP in the list of bound protocols, you will need to remove it.
5. To remove NetBEUI, select the NetBEUI protocol in the list, then click **Remove**.
6. Close the remaining Network dialog windows.
7. Reboot your computer.

After rebooting, the client Ethernet card will use TCP/IP to send and receive network transmissions. You will now be able to use the Windows **Start > Find > Computer** utility to locate the TeamSite server.

You can also use the Advanced tab of the protocol settings to specify TCP/IP as the default protocol binding for Windows Networking, but this solution is not as reliable because it might be upset as network cards are changed and protocols are added and removed.

Configuring PDC

You can configure TeamSite to use the Windows NT Primary Domain Controller (PDC) for name and password authentication. This configuration eliminates the need for users to enter their passwords manually whenever they connect to the TeamSite server.

To use the Windows NT PDC for authentication, modify your Samba configuration as follows:

1. In the [global] section `iw.smb.conf`, modify or add the following:

```
# Select a Windows NT PDC for your password server  
password server = DNS_name_of_your_PDC  
# Use share level security  
security = server  
# Use password encryption  
encrypt passwords = yes
```

2. Stop and restart Samba:

```
% /etc/init.d/iw.samba stop  
% /etc/init.d/iw.samba start
```

NFS Clients

If you are using an NFS client, follow your program's setup instructions to mount `/iwserver` on the TeamSite server as a networked drive. If you can, you should modify the client configuration to *NFS Version 2* and turn off NFS locking (sharing) on the client. If you do not turn off locking, operations might freeze for long periods of time.

FTP Clients

If you are using an FTP client, follow your program's setup instructions to install the software. Log in to the TeamSite server using your UNIX login and password and navigate to your workarea, located at:

```
/iwmnt/default/main/branchpath/WORKAREA/workareaname
```

Macintosh Clients

To use the TeamSite file system interface for Macintosh, you need to have an AppleShare server set up for the server that is running TeamSite as follows:

1. In the Chooser, select **AppleShare**.
2. Select the name of the TeamSite server. Click **OK**.
3. If you are asked for your username and password, enter your TeamSite username and password, and click **OK**.
4. Select the items you want to use, and click **OK**.

The TeamSite server is displayed on your desktop.

For additional AppleShare server recommendations, contact Interwoven Client Services.

UNIX Clients

To access the TeamSite server using UNIX clients, log in to the TeamSite server using your TeamSite username and password.

If you have UNIX clients that will be accessing the TeamSite file system, you can mount the TeamSite directory, or configure the client machine to automatically mount the file system at boot time.

To mount the TeamSite directory, issue the following commands:

```
% mkdir /iwmnt  
% mount -overs=2 servername:/iwserver /iwmnt
```

In the example, *servername* is the name of the server on which TeamSite is running.

Alternatively, you can mount a subdirectory of *iwserver*, for example, a specific branch or workarea. To mount a workarea, issue the following commands:

```
% mkdir /iwmnt  
% mount -overs=2 servername:/iwserver/default/branchpath/WORKAREA/workareaname /iwmnt
```

In the example, *servername* is the name of the server on which TeamSite is running, *branchpath* is the path of the branch your workarea is on (for example, *main/intranet*), and *workareaname* is the name of your workarea.

To set up a UNIX client (including the TeamSite server itself) to mount the TeamSite directory at boot time, edit the */etc/vfstab* file to include the following line:

```
servername:/iwserver - /iwmnt nfs - yes vers=2,bg
```

Loading Content

The TeamSite installation program automatically creates the *main* branch. It contains a staging area and an empty initial edition. Before you start using TeamSite for production, you must transfer your current Web site files into TeamSite.

Populating TeamSite with your content involves the following six general steps. Detailed procedures for each of these steps are in the sections that follow.

1. Create a subbranch for your web developers.
2. Create a TeamSite workarea on the subbranch.
3. Populate the newly created workarea with existing files.
4. Set permissions on the files, or configure a submit filter (see “Submit Filtering” on page 167).
5. Submit the workarea to the staging area.
6. Publish an edition from the staging area.

The newly published edition will then become the foundation of all subsequent work done in TeamSite.

Creating a Subbranch

Interwoven recommends that all development take place on subbranches. The main branch is not usually used for development for several reasons. First, it requires a user with Master privileges to administer. In addition, if you are using TeamSite to develop multiple Web sites, development of one Web site on the main branch and other Web sites on subbranches may create a false hierarchy of branches—the subbranch will not necessarily bear any relation to the parent branch.

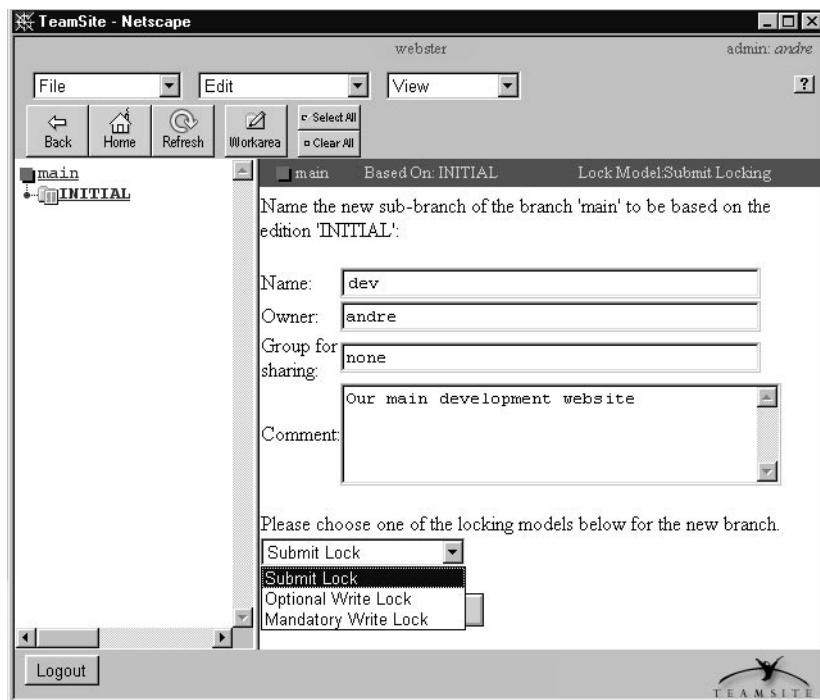
To create a subbranch using the TeamSite GUI:

1. Log in to TeamSite as a Master user.
2. Select **File > New Branch**. Because there is only one edition on the parent branch (the empty INITIAL edition), this subbranch will be based on that edition.
3. A Create Branch window will appear.
4. Enter the name of the branch in the **Name** box. Do not use spaces or the following characters in the branch name:
 \ / : * ? " < > |

Do not name a branch WORKAREA, STAGING, or EDITION.

5. Your username will appear in the **Owner** box. If you want to assign the branch to someone else, type the owner's name in this box.
6. If you want this branch to have multiple Administrators, type the name of the group who will be able to administer this branch in the **Group for Sharing** box. The Administrator or Administrators of this branch will be able to create workareas and subbranches of development. For more information on Administrator privileges, see page 83 and page 91.
7. Use the pull-down menu to select the type of locking you want to be used on this branch (see the *TeamSite User's Guide* for an explanation of the different types of locking).
8. Add any comments in the **Comment** box (comments cannot be changed). Click **OK**.

Your newly created branch will contain no workareas, a staging area, and an empty edition called INITIAL.



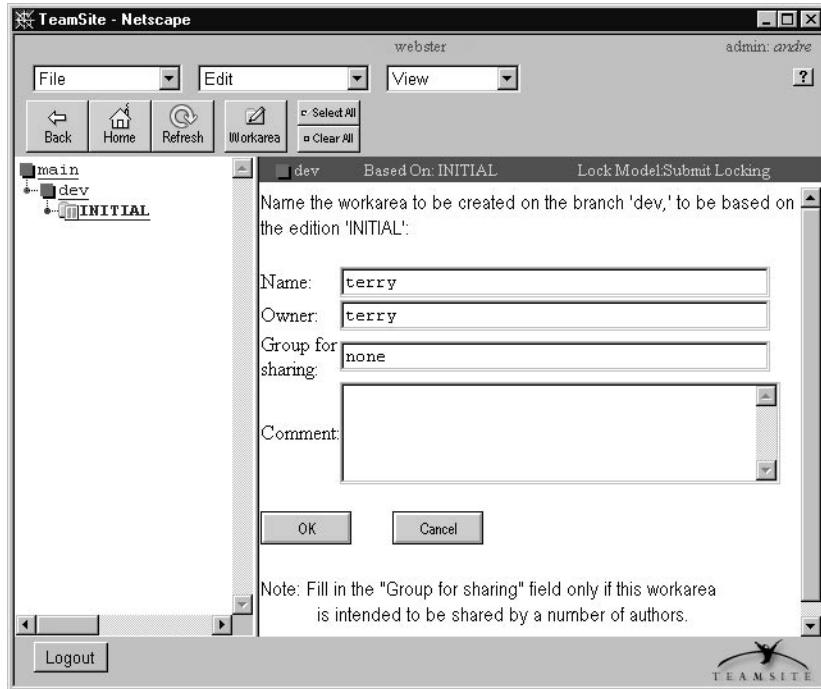
Create Branch window

You can also use the `iwmkbr` command-line tool to create a new branch (see *TeamSite Command-Line Tools*).

Creating a Workarea

To create a workarea using the TeamSite GUI:

1. Click the name of the subbranch you just created, to navigate into the branch.
2. Select **File > New Workarea**. Because there is only an empty edition on this branch, TeamSite will create an empty workarea.
3. The Create Workarea window will appear. Type in the name you want to give the workarea in the Name box, and the username of the workarea's owner in the Owner box.
Avoid using spaces and most punctuation characters in workarea names. Workarea names should consist only of alphanumeric characters, hyphens, and underscores.
4. If you want a group to be able to share this workarea, type the name of the group in the Group for Sharing box. If you want this workarea to be private, so that only the owner can modify files in it, leave the default group (none) selected.
5. Add any comments in the Comments box. Click **OK**.



Create Workarea window

You can also use the `iwmkwa` command-line tool (see *TeamSite Command-Line Tools*).

Populating an Initial Workarea

To populate an initial workarea:

1. From the UNIX file system (via telnet, NFS, Samba, etc.), log in as `root`.

2. Copy all of the original Web site files into the new workarea (default location):

`/iwmnt/default/main/branchname/WORKAREA/workareaname`

where `branchname` is the name of the newly created subbranch and `workareaname` is the name of the newly created workarea on the subbranch.

When copying files, use `tar` to maintain file permissions and timestamps (that is, make a tar file of your Web site content, copy the file into the workarea, then untar the file). When you're done, navigate into the workarea and double-check file permissions before submitting the files to the staging area.

3. Set permissions on the files in your Web site. Use `chown` and `chgrp` to limit access to files by changing the owners and groups for these files. For more information on the `chown` and `chgrp` commands, consult a UNIX reference manual.

Because TeamSite considers a change in permissions to be a change in the file, TeamSite will store a new version of the file when you change its permissions (new versions are created at the time files are submitted to the staging area). If you wait to set permissions until after your files have been imported into a workarea and submitted to the staging area, you can create a large number of extra versions and unnecessarily clutter each file's version history. To avoid creating unnecessary versions, set permissions immediately after you populate the workarea (but before you submit the files). Interwoven recommends that you configure a submit filter to automate this process (see page 167), but you can also set permissions manually.

Note: Be sure to set permissions **before** you submit files to the staging area for the first time.

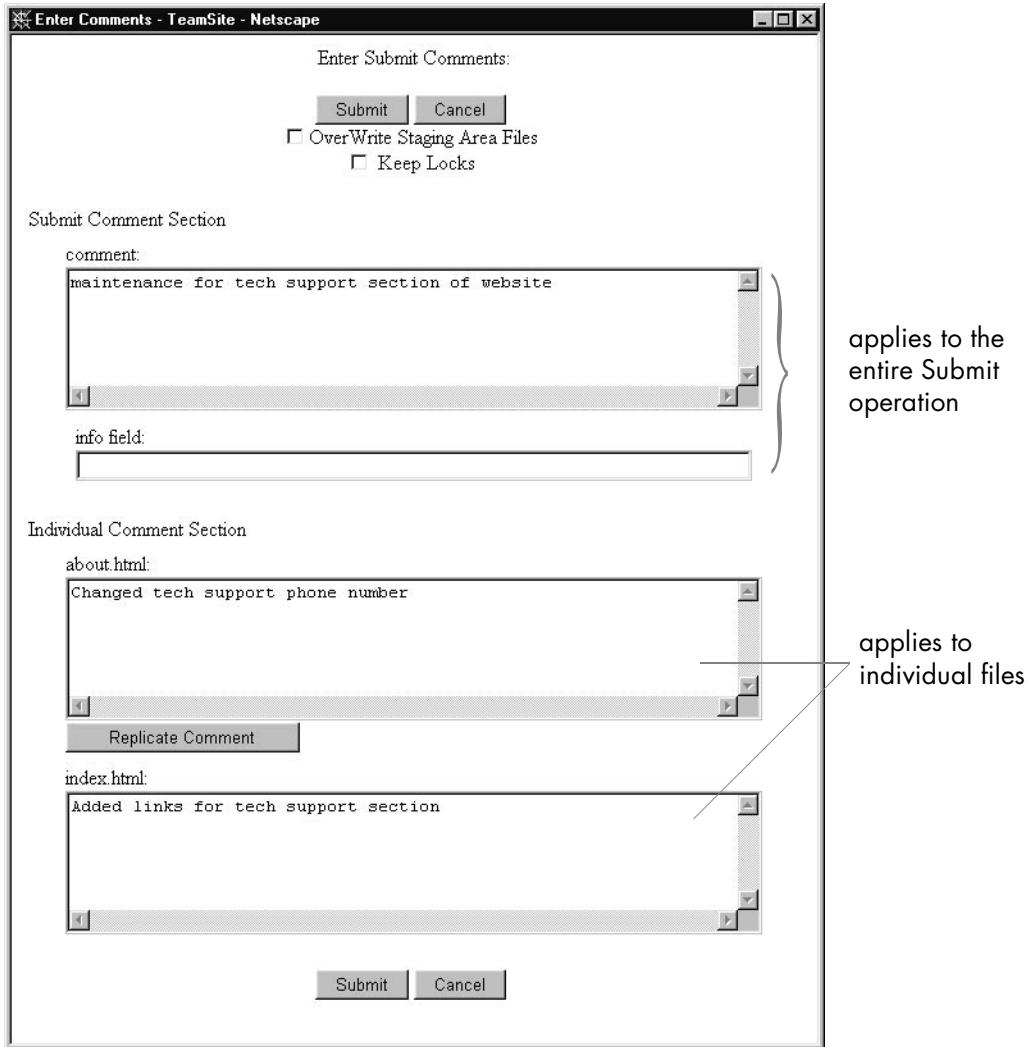
Submitting Files to the Staging Area

Now that you have populated your workarea with your Web site content, you can submit it to the staging area. You need to submit your content to the staging area before you can publish an edition, which you can use as the basis of all future workareas.

To submit the contents of your workarea to the staging area via the TeamSite GUI:

1. Go to the top level of the workarea. Do not select any checkboxes.
2. Select **File > Submit Direct**.¹ A dialog box will appear asking if you want to submit the entire directory.
3. Click **OK**. A Submit window will appear.
4. Enter any comments you have in the comment boxes. The Submit window contains two sections:
 - The **Submit Comments** section, which consists of a section where you can enter comments for the entire Submit operation, and a field where you can enter keywords, for example, for automatic triggers.
 - The **Individual Comments** section, where you can attach comments to each file.
5. Click the **Submit** button.

1. For first-time submissions of large numbers of files, you should use the direct Submit option rather than workflow Submit.



Submit Comments window

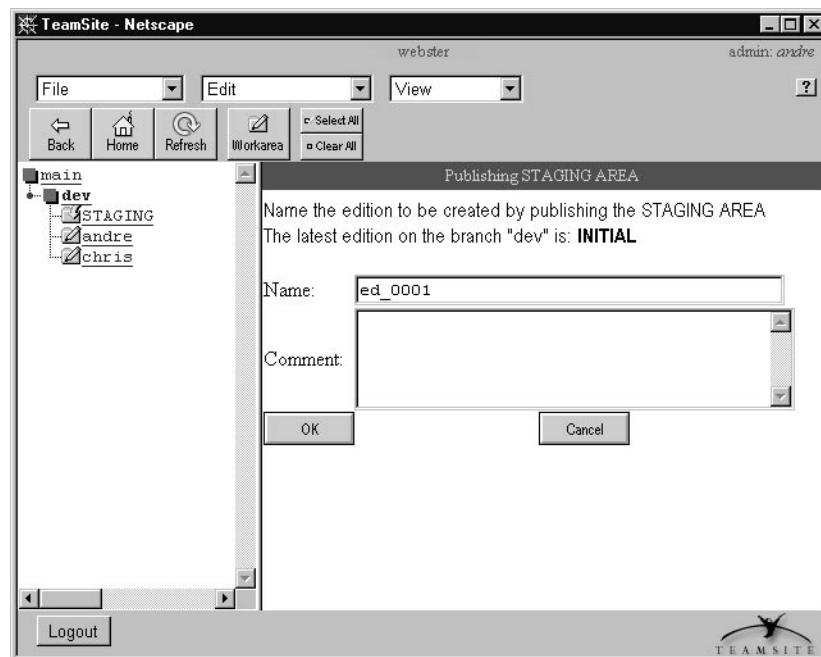
You can also use the `iwsubmit` command-line tool to submit files to the staging area (see *TeamSite Command-Line Tools*).

Publishing a New Edition

Publishing an edition creates a snapshot of the staging area at the time of publication. These editions can be used as checkpoints. As part of your initial installation process, you should create an edition to record the state of your Web site at the time that you installed TeamSite. You can use this new edition as the basis for the other workareas you create on this branch.

To create a new edition from the contents of the staging area via the GUI:

1. Select **File > Publish** menu from anywhere within your branch to display the Publish window.
2. TeamSite will suggest a name for the new edition. If you want to give the edition a different name, enter the name of the new edition in the **Name** box in the Publish window.



Publish window

3. Enter any comments you have in the **Comments** box.
4. Click **OK**. The staging area will be published as a new edition.

You can also use the `iwpublish` command-line tool to publish an edition (see *TeamSite Command-Line Tools*).

Uninstalling TeamSite

If you are currently running TeamSite release 5.0.1 with Service Pack 1, you must uninstall the Service Pack before uninstalling TeamSite. To uninstall TeamSite 5.0.1 Service Pack 1:

1. Navigate to the directory that contains the `uninstallsp.sh` file.

By default it is located in a directory called `IWOV_Upgrades/TeamSite/5.0.1/1.0.8604` located at the same level as `iw-home` (you can run `iwgethome` to have TeamSite return this information) .

2. Start the `uninstallsp.sh` file:

```
% ./uninstallsp.sh
```

3. Respond to the prompts to complete the uninstallation procedure.

To uninstall TeamSite complete the following procedure:

1. Log onto your system as **root**.

2. Find the TeamSite installation directory, `iw-home`, by running the following command:

```
% iwgethome
```

3. Run the following command:

```
% iw-home/install/iwuninstall
```

Use this command with extreme caution! It will remove all traces of TeamSite from the host machine. Once triggered, this command cannot be reversed except by restoring files from your backups.

The **iwui** User

The TeamSite installation program creates a UI daemon user called **iwui** to restrict access to the session cookie encryption password stored in **iw-home/private/etc/passphrase**. The UI daemons (**iwwebd** and **iwservletd**) and **iproxy** must be able to read the password to issue and validate session cookies. Currently, **iwwebd** and **iwservletd** run as **iwui**, and **iproxy** runs as **root**. The **iwui** user has no privileges except for being able to read the **passphrase** file:

```
-r----- 1 iwui adm 94 Aug 1 16:20 /usr/iw-home/private/etc/passphrase
```

The **iwui** user also has the following characteristics:

- Member of the system group **nobody**—This ensures there are as few privileges associated with this user as possible. Membership in other groups which own no files on the system would be equivalent.
- Owns the **iwwebd** and **iwservletd** processes—For more information about **iwwebd** see page 235, for more information about **iwservletd** see page 234.
- Is hardcoded as **iwui** in the **iwsessionkeygen** CLT—Therefore you should not change the name of this user.
- Impersonation user for Tomcat—Provides the ability for the user logged into the browser interface to execute a program on the TeamSite server and have that program behave as if it were run by the logged in user and not **iwui**, **nobody**, or **root**.
- By default, is assigned the next available user ID (UID) by the installation program:
*Creating Interwoven TeamSite UI Daemons User (iwui)...
Please choose a UID for the new user, or press <Enter> to accept the next available UID.*
- Home directory is **iw-home**

If you do not want this system user to have a UID mixed in with your regular user accounts, manually assign a UID during the installation (system accounts typically have a lower UID).

Chapter 3

Managing Access

Security

Access to TeamSite is governed by the following two factors:

- UNIX-related permissions
- TeamSite access privileges

UNIX file permissions control who has access to individual files and directories. UNIX password authentication is used when logging in to TeamSite. However, TeamSite access privileges govern who can log in under various roles, and who has access to branches and workareas. For example, to edit a file in a workarea, a user must both be able to access that workarea (through TeamSite access privileges), and have permissions for that file and its parent directory (through UNIX permissions). For a complete list of the TeamSite and UNIX permissions needed to perform any action in TeamSite, see page 91.

When adding a new user, you need to consider the following three factors:

- Whether the user will have access to the server.
- The role the user will play in your Web site operations.
- The portion of the Web site the user will be editing.

If the user does not have access to the TeamSite server, he or she will need to be added as described on page 84. To decide what TeamSite role best corresponds to the new user's role in your Web site operations, see page 83.

To decide what groups the new user needs to belong to, and which workareas he or she needs to access, consider your existing groups and which portions of the Web site and which workareas they can access. Add the new user to the groups that work on the same portion of the Web site that he will be editing, and he will automatically have access to their workareas and to their Web site files. If the new user needs his or her own workarea, create a private or shared workarea, but make sure that he or she owns or has group-level access to the files that he or she will be editing. To change ownership or group access of files, see page 86.

When creating a new workarea, you need to decide:

- What the name of the workarea should be.
- Who will need to access the workarea (this can be one person, or one person and a group).
- What portions of the Web site the workarea's owner and group should and should not have access to.

Set permissions on your files according to the latter consideration. Remember that permissions cannot be set differently for different workareas. If the permissions for corresponding files are set differently in different workareas, you will encounter conflicts when you submit files to the staging area.

It is often useful to keep a chart of your Web site that shows what users and groups have access to what sections of the Web site.

Note: The TeamSite GUI accepts passwords up to 13 characters, but other underlying authentication operating system mechanisms (including /etc/passwd, PAM, LDAP, and SecureID) may have different policies. The UNIX default password is a maximum of eight characters.

Users

TeamSite Roles Overview

To facilitate workflow and security, TeamSite provides users with four roles, each with varying levels of access to TeamSite: Master, Administrator, Editor, and Author. To determine which role a user should have, consult the following table and find the role that best fits the user's work.

Author	Editor	Administrator	Master
Owes content	Owes workareas	Owes branches	Owes main branch
Edits & creates files	Edits & creates files	Edits & creates files	Edits & creates files
Receives assignments	Assigns files	Assigns files	Assigns files
Work is approved by workarea owner	Approves/rejects work of Authors	Approves/rejects work of Authors	Approves/rejects work of Authors
	Uses advanced version management features	Uses advanced version management features	Uses advanced version management features
	Maintains content of workarea	Manages branch	Manages entire Web site
	Submits files to the staging area	Submits files to the staging area	Submits files to the staging area
	Publishes editions (optional)	Publishes editions	Publishes editions
		Creates and deletes workareas	Creates and deletes workareas
		Creates and deletes sub-branches	Creates and deletes sub-branches

In addition, Administrators can perform all the functions that Editors can. Master users can perform all the functions that Administrators and Editors can.

Adding and Removing Users

Adding Users

Before you can add a user to TeamSite, he must be a UNIX user on the TeamSite server. Always consult your system administrator before adding a user. If the user already exists, skip to Step 4.

To add a user to TeamSite:

1. Add the user to the UNIX password file for the TeamSite server. **Always consult your system administrator before adding a user to this file.** Your system administrator may want to add the new user himself. For more information, consult a UNIX system administration reference.
2. Add the user's UNIX login name to the appropriate TeamSite roles file(s).

The four TeamSite roles files are in the directory `iw-home/conf/roles`. This directory may also contain roles files for other Interwoven products.

`master.uid`
`admin.uid`
`editor.uid`
`author.uid`

Each file contains a list of the users who have privileges for that role, one user to a line.



The screenshot shows a Telnet session titled "Telnet - athena". The menu bar includes "Connect", "Edit", "Terminal", and "Help". The command entered is "sol:teamsite:/private/iw-home/conf/roles:118% more editor.uid". The output lists several user names:

```
root
test1
test2
test3
test4
guestacc
dialins
andre
chris
chrisc
bobbie
pat
nobody
--More--(11%)
```

An `editor.uid` file

To give a user multiple roles, include his name in multiple .uid files. For example, an Editor may also be able to log in as an Author. A Master user should be able to log in with any role, so you will need to include the name of each Master user in each .uid file. TeamSite users' passwords are the same as their UNIX password.

After you have edited the TeamSite roles files, you will need to tell TeamSite to reread them. type:

```
% iwreset
```

The TeamSite server will return 0 on success, non-zero on failure.

At this point the new user will now be able to log in to TeamSite, but he will not have access to any branches or workareas.

3. Add the user to the appropriate UNIX groups files. If you want the user to have access to a shared workarea, add him to the group that has access to that workarea. If the user is an Administrator, and you want him to have Administrator privileges for a branch, add him to the group of Administrators for that branch.

To add a user to a group, edit the /etc/group file. Locate the name of the group you want to add the user to, and add his name to the list of usernames that follows it. Usernames must be separated by commas.

4. If you want the user to own a workarea, create a workarea for him on the sub-branch where he will be working (see the *TeamSite User's Guide*).

Deleting Users

To remove a user from TeamSite:

1. If your installation stores TeamSite role information in .uid files, delete the user's name from each of these files. If your installation uses LDAP to store role information, delete all the TeamSite roles from the user's LDAP entry.
2. Use **iwreset** to cause TeamSite to reread the user information from the roles files or LDAP database (see *TeamSite Command-Line Tools*).
3. Remove the user from TeamSite's entity database:

```
% iwuser -d username
```

where *username* is the username of the user you want to remove.

If you do not perform this step, you will not be able to create another user with the same name.

You might also want to remove the user's UNIX user account for the TeamSite server:

1. Delete his username from any groups that he belongs to.
2. Remove him from the `/etc/passwd` file. Always consult your system administrator before altering the `/etc/passwd` file in any way.

Access Control

To control access to individual files or directories, use `chmod` to change the permission bits. Use `chown` to change the ownership of files or directories, and `chgrp` to change the group. For more information on `chmod`, `chown`, and `chgrp`, consult a UNIX reference manual.

You may also want to use a submit filter (see "Submit Filtering" on page 167) to automatically change and enforce permissions on files or directories.

Note: `chmod`, `chown`, and `chgrp` commands, when used recursively, touch every file in the directory, whether they need to or not. This may generate excess TeamSite versions. Instead, target the command using `find`:

```
% find . ! -group webedit -exec chgrp webedit {} \;
% find . ! -perm -g+w -exec chmod g+w {} \;
```

This will touch only the necessary files and not generate unnecessary versions.

Group Membership

Many workareas are shared by groups. For a user to have access to a particular workarea, he must either be the owner of the workarea, or a member of the workarea's group. TeamSite uses UNIX groups for access control. These groups can be managed with standard UNIX commands. To add a user to a group, edit the `/etc/group` file. Locate the name of the group you want to add the user to, and add his name to the list of usernames that follows it. Usernames must be separated by commas.

To create a group, add a new line to the `/etc/group` file in the following format:

`groupname:*:gid:username1,username2,username3`

For example, the entry for the group `allauthors` might look like this:

`allauthors:*:2000:pat,andre,chris`

You can add as many users to a group as you want.

Changing Group Ownership of Workareas

To change which group has access to a workarea:

1. Navigate into the directory containing the workarea.
2. Use the `chgrp` command to change the workarea's group. For more information on the `chgrp` command, consult a UNIX reference manual.

Example

In this example, user Chris changes the group for one of the workareas on the main branch. First, he navigates into the directory containing the workareas. Then, he looks at the existing workareas and learns that Andre has two workareas, one private and one shared with the group `demoauthor`. Chris has one private workarea, `wa1`. He uses the `chgrp` command to change the group on his workarea, then checks the results.

```
% cd /iwmnt/default/main/WORKAREA
% ls -la
total 3
drwxrwxr-x  27 andre      nobody          512 Apr 23 17:07 andre1
drwxrwxr-x   3 andre      demoauthor    512 Apr 17 11:52 andre2
drwxrwxr-x   2 chris     nobody          512 Apr 17 11:37 wa1
% chgrp demoauthor wa1
% ls -la
total 3
drwxrwxr-x  27 andre      nobody          512 Apr 23 17:07 andre1
drwxrwxr-x   3 andre      demoauthor    512 Apr 17 11:52 andre2
```



```
drwxrwxr-x    2 chris      demoauthor    512 Apr 17 11:37 wa1
```

Checking User Roles

The TeamSite Command Line Tool `iwckrole` allows you to check whether or not a user can log in with a certain role.

Usage

```
iwckrole [-h|-v] role user
```

<code>-h</code>	Displays usage message.
<code>-v</code>	Displays version.
<i>role</i>	<code>author</code> , <code>editor</code> , <code>admin</code> or <code>master</code> .
<i>user</i>	Username of the person whose role you are checking.

Exits with YES on successful authorization, NO on failure.

Example

```
% iwckrole admin andre
```

returns:

YES

indicating that user “andre” can log in as an Administrator.

Locking Models

TeamSite supports three different types of file locking:

- Submit Locking
- Optional Write Locking
- Mandatory Write Locking

A branch's locking model is set when it is created (for more information, see the *TeamSite User's Guide*). Different branches on one TeamSite server may use different types of locking. All workareas on a branch use the same type of locking.

When a file is locked, it is locked for a particular workarea. That is, all users who have access to that workarea can edit the file. In addition, all users who have previously modified the file can edit it in their workareas (but not lock it).

Submit Locking

Submit locking means that if a file is locked, only users within the workarea where it is locked may submit the file to the staging area. Users are still allowed to edit the file within the context of other workareas but may not submit it until the user who holds the lock has submitted his version or manually released the lock.

If a file is not locked, anyone may submit it.

If someone else has the lock on a file that a user is editing, and the user tries to submit a workarea or directory containing his version after the lock holder has submitted the file and released the lock, the Compare Results window will appear showing the conflicting versions of the file. From this window, the user can choose to merge the two files, or to overwrite the version in the staging area with his own.

If someone else has locked a file, and a user edits it through the TeamSite GUI, the following warning is displayed:



The user can continue to edit the file, but will have to merge the changes with those of the lock owner after it is submitted.

Write Locking

Write locking means that a locked file may only be edited in the workarea where it is locked. Users in other workareas may not edit the same file even within the context of their own workareas, but they may view a read-only copy if they have the necessary permissions. Write locking may be optional, in which case users may choose whether or not to lock the files that they edit, or mandatory, in which case users cannot edit a file without locking it first. Under Mandatory Write Locking, all files in a workarea are read-only until a user locks them for editing. Once a user modifies a file while holding a lock, the user will be able to continue to modify the file even after releasing the lock. Once the user submits the file, it will become read-only again.

If a user edits an unlocked file, and somebody edits the file at the same time but in a different workarea, the second person to submit the file to the staging area will have a conflict and will need to either merge the two versions, overwrite the version in the staging area with his own, or overwrite his own version with the version in the staging area (see the *TeamSite User's Guide* for details).

Permissions

When a user tries to perform any action in TeamSite, the TeamSite server automatically checks to see whether or not he has permission to perform that action. TeamSite checks the following factors:

- User roles
- Branch permissions
- Workarea permissions

- File permissions
- Directory permissions

Not all of these factors apply to every action. TeamSite only checks the factors that apply to the action being attempted.

The table below lists which privileges a user must have in order to perform any action in TeamSite. To find out whether a user will be able to perform a specific action, check the entry for that action under the user's role and determine whether or not the specified conditions apply. All conditions listed in each box below must apply in order for a user to perform an action, unless otherwise specified.

Note that TeamSite workflow tasks may require users to perform actions such as editing a file or submitting it to the staging area. To perform the task, the user must have the ability to perform the action as specified in the table below. For example, if you assign a task that requires an Author to edit a file, the Author must have workarea permissions, parent directory permissions, and file permissions for that file as specified in the table beginning on page 93.

User roles: If you are attempting to perform any of these actions through the GUI, you must be logged in with the role specified. If you are using the file system interface, you must be able to log in with the specified role.

Branch permissions: A user has branch permissions if he is either the primary owner of the branch or a parent branch, or if he belongs to the group that owns the branch or a parent branch. Master users automatically have branch permissions for all branches. Only Administrators and Master users can have branch permissions.

Workarea permissions: A user has workarea permissions if he is either the primary owner of a workarea, or if he belongs to the group that has access to the workarea. Workarea permissions are usually synonymous with read-write-access to the root directory of the workarea—the default setting for workarea permissions is 775. If different permissions are specified for the owner and group, some sections of the table below will not apply. If “world” is given read-write-execute permissions, then all users will be considered members of the workarea for those conditions indicated with asterisks (*).

File permissions: File permissions are UNIX read-write-execute permissions (unless otherwise specified) to a file.

Directory permissions: Directory permissions are UNIX read-write permissions (unless otherwise specified) to a directory.

Task ownership: Includes the ability to take ownership of a task, for group tasks.

	Author	Editor	Administrator	Master
Edit file ¹	workarea permissions* parent directory permissions (read/ execute) file permissions (write)	workarea permissions* parent directory permissions (read/ execute) file permissions (write)	workarea permissions* parent directory permissions (read/execute) file permissions (write)	workarea permissions* parent directory permissions (read/execute) file permissions (write)
View file	workarea permissions* parent directory permissions (read/ execute) file permissions (read)	workarea permissions* parent directory permissions (read/ execute) file permissions (read)	workarea permissions* parent directory permissions (read/execute) file permissions (read)	workarea permissions* parent directory permissions (read/execute) file permissions (read)
New file ²	workarea permissions* parent directory permissions (read/ write/execute)	workarea permissions* parent directory permissions (read/ write/execute)	workarea permissions* parent directory permissions (read/write/ execute)	workarea permissions* parent directory permissions (read/write/ execute)
Move file Rename file ³	workarea permissions* parent directory permissions (read/ write/execute)	workarea permissions* parent directory permissions (read/ write/execute)	workarea permissions* parent directory permissions (read/write/ execute) OR branch permissions	Yes

	Author	Editor	Administrator	Master
New directory	workarea permissions* parent directory permissions (read/write/execute)	workarea permissions* parent directory permissions (read/write/execute)	workarea permissions* parent directory permissions (read/write/execute)	workarea permissions* parent directory permissions (read/write/execute)
Move directory Rename directory	workarea permissions* parent directory permissions (read/write/execute)	workarea permissions* parent directory permissions (read/write/execute)	workarea permissions* parent directory permissions (read/write/execute) OR branch permissions	Yes
Delete file ⁴	No ⁵	workarea permissions* parent directory permissions (read/write/execute)	workarea permissions* parent directory permissions (read/write/execute) OR branch permissions	Yes
Delete directory	No	workarea permissions* parent directory permissions (read/write/execute)	workarea permissions* parent directory permissions (read/write/execute) OR branch permissions	Yes

	Author	Editor	Administrator	Master
Copy ⁶ (through the TeamSite GUI)	file permissions (read) directory permissions (destination directory) workarea permissions	file permissions (read) directory permissions (destination directory) workarea permissions	file permissions (read) directory permissions (destination directory) workarea permissions	file permissions (read) directory permissions (destination directory) workarea permissions
Lock file ⁷	No ⁸	workarea permissions parent directory permissions (read/execute) file permissions (write or ownership)	workarea permissions parent directory permissions (read/execute) file permissions (write or ownership)	workarea permissions parent directory permissions (read/execute) file permissions (write or ownership)
Unlock file	No	creator of lock OR owner of workarea	creator of lock OR owner of workarea OR branch permissions	Yes
Revert	No	workarea permissions	workarea permissions OR branch permissions	Yes
Get Latest	No	workarea permissions	workarea permissions OR branch permissions	Yes

	Author	Editor	Administrator	Master
Copy To	No	workarea permissions (destination workarea)	workarea permissions (destination workarea) OR branch permissions (destination)	Yes
Set Public/ Private	No	workarea permissions	workarea permissions OR branch permissions	Yes
View History	No	workarea permissions (read)	workarea permissions (read)	Yes
Compare	No	workarea permissions (read)	workarea permissions (read)	Yes
List Modified	No	workarea permissions (read)	workarea permissions (read)	Yes
List Locks	No	workarea permissions (read)	workarea permissions (read)	Yes
View Submit Log	No	workarea permissions (read)	workarea permissions (read)	Yes
View Update Log	No	workarea permissions (read)	workarea permissions (read)	Yes
Create branch	No	No	branch permissions	Yes
Delete branch	No	No	branch permissions	Yes
Rename branch	No	No	branch permissions	Yes
Submit files	No ⁹	workarea permissions	workarea permissions OR branch permissions	Yes

	Author	Editor	Administrator	Master
Publish edition	No	workarea permissions for any workarea on the branch	workarea permissions for any workarea on the branch OR branch permissions	Yes
Delete edition	No	No	branch permissions	Yes
Rename edition	No	No	branch permissions	Yes
Create workarea	No	No	branch permissions	Yes
Delete workarea	No	No	branch permissions	Yes
Rename workarea	No	No	branch permissions	Yes
View reports	No	No	Yes	Yes
Assign file	No	workarea permissions	workarea OR branch permissions	Yes
View task	task ownership	task ownership OR job ownership	task ownership OR job ownership	task ownership OR job ownership
View job information	ownership of a task within the job	job ownership ownership of a task within the job	job ownership ownership of a task within the job	job ownership ownership of a task within the job
Create new job	No	available_tempplates.cfg setup	available_tempplates.cfg setup	available_tempplates.cfg setup
Transition task	job or task ownership	job or task ownership	job or task ownership	job or task ownership
Add/remove files from existing task	job or task ownership	job or task ownership	job or task ownership	job or task ownership
View task changes	job or task ownership	job or task ownership	job or task ownership	job or task ownership

	Author	Editor	Administrator	Master
Compare task files (with staging area)	job or task ownership			
Revert task files (to staging area version)	job or task ownership			

1. The ability to edit a file only applies to files that are not already write-locked. If the file is write-locked, then only the lock owner can edit it. Note that if an Author edits a file through the GUI, TeamSite will lock the file.
2. The ability to create a file only applies to files that are not already write-locked. You cannot create a file with the same name as a file that is already write-locked. If an Author creates a file, the new file will be assigned to him.
3. The ability to rename or move a file only applies to files that are not already write-locked. If the file is write-locked, then only the lock owner can rename it. A file cannot be renamed with the name of a file that is locked. If an Author renames or moves a file, the renamed or moved version of the file will be assigned to him.
4. For Authors and Editors, the ability to delete a file only applies to files that are not already write-locked. If the file is write-locked, then only the lock owner can delete it.
5. Authors can delete files through WebDesk only.
6. If an Author copies a file, the copied version of the file will be assigned to him.
7. The ability to lock a file only applies to files that are not already locked.
8. Authors can lock files through WebDesk only.
9. Authors cannot submit files directly. All work done by Authors must go through an approval process prior to submission. The approver must have the ability to submit files.

Chapter 4

Using the Interwoven Administration GUI

TeamSite includes a graphical user interface (GUI) framework that enables users to perform administrative tasks across a variety of Interwoven products. The Interwoven Administration GUI is a Web application accessible from any system with a compatible browser (Netscape Navigator or Internet Explorer).

This chapter is limited to the TeamSite portion of the Interwoven Administration GUI. For information on managing the settings for other Interwoven products through the Interwoven Administration GUI, consult the documentation for those products.

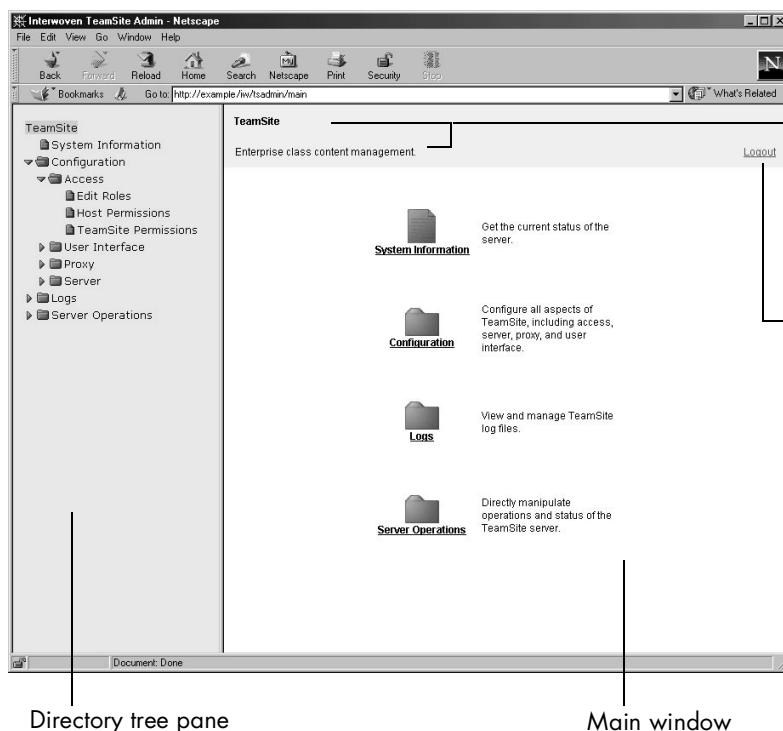
The TeamSite section of the Interwoven Administration GUI provides an easy-to-use interface to:

- **View system information**—View basic information such as:
 - Server status
 - Version of the TeamSite server
 - License expiration date
 - Disk space
 - Load and throughput data
 - Server operations and the users who performed them
- **Configure the TeamSite client interface, proxy, and server**—Configure TeamSite without having to manually edit the main TeamSite configuration file, `iw.cfg`. When you apply a change to a setting through the GUI, the corresponding section in `iw.cfg` is edited. The TeamSite server automatically executes the changes when it next polls `iw.cfg` (usually within one minute).

When you apply changes through the GUI, the order of the options in `iw.cfg` and any comments therein are preserved. Additionally, a warning is displayed when you attempt to change a setting if any part of the `iw.cfg` file has been modified (either manually or through the GUI) between the time you loaded the current page and when you attempt to make the change. See “Apply, Refresh, and Cancel” on page 101 for details on responses to the warning.

Note: The `iw.cfg` file is owned by the user `iwui`. Do *not* change the ownership of this file.
For more information about `iwui`, see page 80.

- **View and configure log files**—View and configure these TeamSite log files:
 - `iwserver.log`
 - `iwevent.log`
 - `iwtrace.log`
- **Perform server operations**—Perform these operations on the TeamSite server without having to manually invoke command line tools (CLTs):
 - Abort
 - Freeze
 - Reset



Window title with navigation trail beneath
(In the initial TeamSite window, a tagline replaces the navigation trail)

Logs you out of the GUI and returns you to the Login screen

The initial window of the TeamSite section of the Interwoven Administration GUI

Users who want to configure TeamSite by directly editing configuration files can find instructions for doing so throughout this book. Where appropriate, this chapter cross-references the more detailed information in other chapters regarding some configuration items.

Navigation

There are three ways to access the settings you want to change:

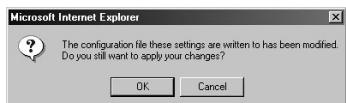
- Use the directory tree in the left-hand pane.
- Navigate through the folders in the main window.
- Use the links in the navigation trail under the window title.

Apply, Refresh, and Cancel

Most of the GUI windows have **Apply**, **Refresh**, and **Cancel** buttons near the bottom.

- **Apply**—Becomes active only when you change a setting. Click **Apply** to write your changes to `iw.cfg`.

If any part of the `iw.cfg` file has been modified between the last time you loaded your current page and when you click **Apply**, a warning dialog box is displayed.



Multiuser Warning

The warning is displayed whether or not the modified section is the same as the section you attempt to change. For example, user A logs in to the GUI and navigates to the Edit Roles. User A begins to enter data to add a user to TeamSite. At the same time, user B (either through the GUI or manually) changes a setting in the proxy section of `iw.cfg`. User A clicks **Apply**. The warning dialog box is displayed even though user A's change will modify a section of `iw.cfg` different than the section changed by user B.

Respond to the warning by doing one of the following:

- Click **OK** to apply your changes.
- Click **Cancel** to abort the process and exit the dialog box.

- Click the **Refresh** button in the GUI to abort the process and reload the page to display the most current settings.
- **Refresh**—Reloads the window so that the most current settings are displayed. In some cases (Edit Roles, for example) clicking **Refresh** clears all input fields.
- **Cancel**—Loads the enclosing title page of the window from which you canceled (for example, if you navigate to the Edit Roles window and click **Cancel**, you are taken to the Access page.) Changes are discarded.

Logging In To the Interwoven Administration GUI

You must have Master permissions on TeamSite to log in to the Interwoven Administration GUI.

To log in to the GUI:

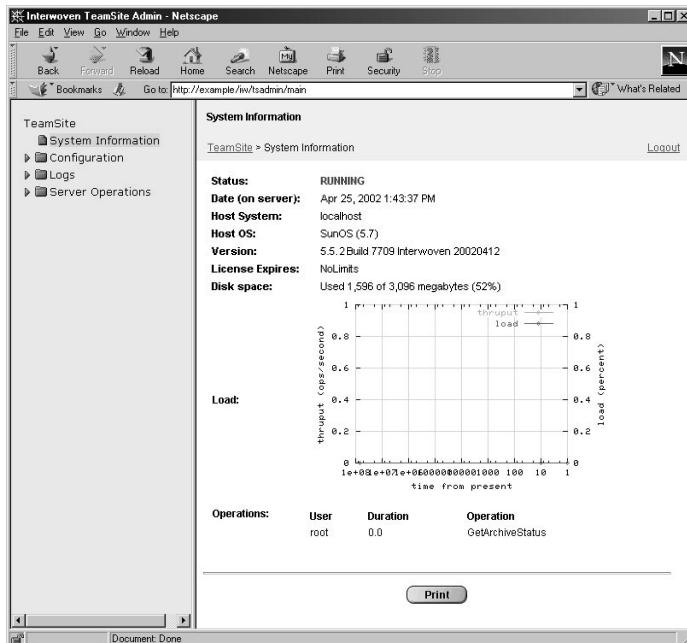
1. Access the Administration GUI by typing the following in your browser:
`http://hostname/iw/tsadmin`
2. Enter your TeamSite user name.
3. Enter your TeamSite password.
4. Click **Login**.

Viewing System Information

In the System Information window you can view:

- **Status**—Displays one of the following states:
 - **Running**—Any backing store is up or frozen.
 - **Stopped**—TeamSite is installed, but nothing is running.
 - **Unknown**—`tsadmin` could not determine the status.
- **Date (on server)**—Displays the date and time (as set on the server).
- **Host System**—Displays the name of the host system.

- **Host OS**—Displays the operating system of the host system.
- **Version**—Displays the version and build number of the TeamSite server.
- **License Expires**—Displays the license expiration date. For details about licenses, see page 47.
- **Disk space**—Displays the amount of disk space used on the host system.
- **Load**—Graphs average load and throughput against minutes of uptime. Data is supplied by the `iwstat` CLT.
- **Operations**—Lists the name, user, and duration (in seconds) of all active server operations.



System Information window

The System Information window is refreshed every 60 seconds.

To print system information, click **Print**.

Editing Roles

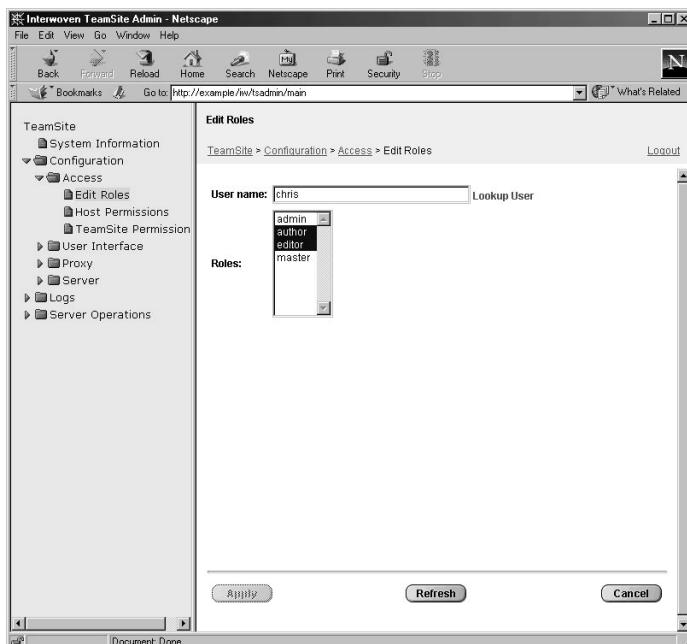
In the Edit Roles window you can:

- Add and remove TeamSite users.
- Edit the roles of existing TeamSite users.

The administration GUI ensures that the user name you enter represents a valid host system user, then displays a list of the TeamSite roles that the user can have.

To add a user to TeamSite, or to edit an existing TeamSite user's roles:

1. Navigate to **TeamSite > Configuration > Access > Edit Roles**.



Edit Roles window, displaying a sample TeamSite user

2. Enter the user name and click **Lookup User**.

Note: If the user is not on the host system, a message is displayed indicating this fact. You must add the user to the host system before you can add the user to TeamSite. To add the user to TeamSite, first add the user to the host system (consult your system's manual), then repeat steps 1-4 in this section.

3. Select the roles you want to give the user. (Hold down the Shift key on your keyboard to make continuous multiple selections. Hold down the Ctrl key on your keyboard to make discontinuous multiple selections.)

4. Click **Apply**.

The user is added to TeamSite with the specified roles.

To remove a user from TeamSite:

1. Enter the user name and click **Lookup User**.

2. Deselect all roles. (Hold down the Ctrl key on your keyboard to deselect the user's last remaining role.)

3. Click **Apply**.

A prompt is displayed asking you to confirm the removal of the user from TeamSite.

4. Click **OK** to remove the user from TeamSite.

If you inadvertently remove yourself from the Master role, you are immediately logged out of the GUI and can log back in only when another Master adds you back to the Master role.

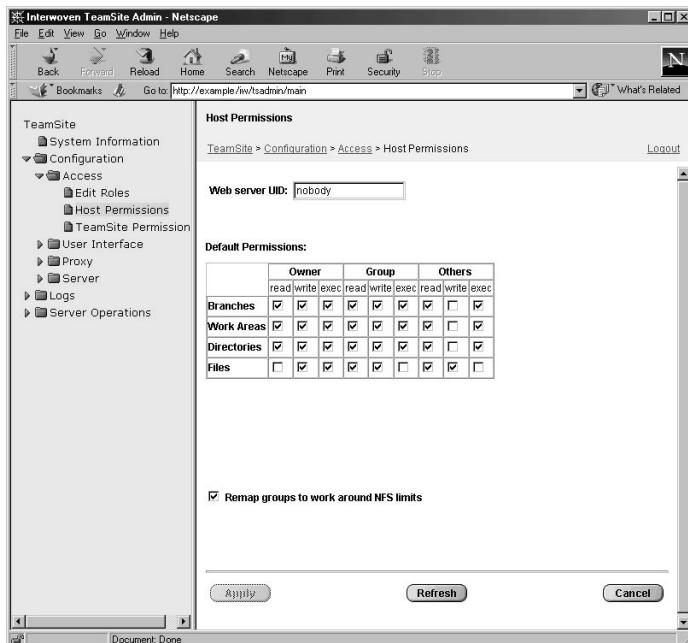
To add or remove a TeamSite user manually, see page 84.

Setting Host Permissions

In the Host Permissions window you can:

- Specify the Web server uid.
- Specify default permissions for branches, workareas, directories, and files created using the TeamSite GUI.
- Activate TeamSite's workaround for NFS limits on number of groups

To configure host permissions, navigate to **TeamSite > Configuration > Access > Host Permissions**.



Host Permissions window

- **Web server UID**—Specifies the Web server uid. You must enter the uid of the Web server that communicates with TeamSite. Entering a different uid here does not change the Web server's uid. A Web server runs as a particular user, usually *nobody*. In order for browsers to view Web content, TeamSite needs this setting to match the uid of the Web server. Because external browsers access the Web server as *nobody*, this is used as the default.
- **Default Permissions**—Specifies default permissions. Permissions on files created through the file system interface are determined by your file system interface configuration (for example, the Samba configuration).

To configure these permissions manually, see page 155.

- **Remap groups to work around NFS limits**—Overcomes NFS group-checking limitations. Because TeamSite does not have the NFS 16-group limitation, it first determines whether a user should have group-level access to the file. Then, if the **Remap groups to work around NFS limits** option is turned on, it maps the file's group to the user's primary group.

Note: When using this option, checking the group ID (GID) on a file through the file system could return a GID different from the true GID of the file. To find the file's true GID, use the **File > File Properties** menu item in the TeamSite GUI, or the **iwattrib** command-line tool.

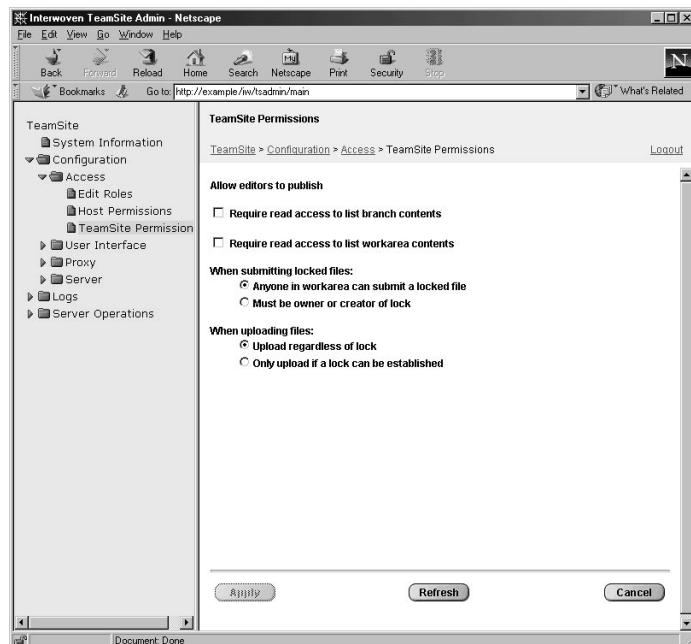
To configure this option manually, see page 156.

Setting TeamSite Permissions

You can specify TeamSite permissions for:

- Whether Editors can publish.
- Whether users can see the names of branches and workareas where they do not have access.
- Whether anyone, or only the owner or creator of the lock, can submit locked files.
- Whether you want files uploaded only if TeamSite can establish a lock.

To modify TeamSite permissions, navigate to **TeamSite > Configuration > Access > TeamSite Permissions**.



TeamSite Permissions window, displaying the default settings

- **Allow editors to publish**—Applies to all Editors on all branches.
To configure this option manually in `iw.cfg`, see page 128.
- **Require read access to list branch contents**—Specifies whether users can see the names of branches where they do not have access.
- **Require read access to list workarea contents**—Specifies whether users can see the names of workareas where they do not have access.

Note: If unchecked, all branch names and workareas appear in the TeamSite GUI, although the branches and workareas where the user does not have read access are not hyperlinked.

To configure these options manually in `iw.cfg`, see page 154.

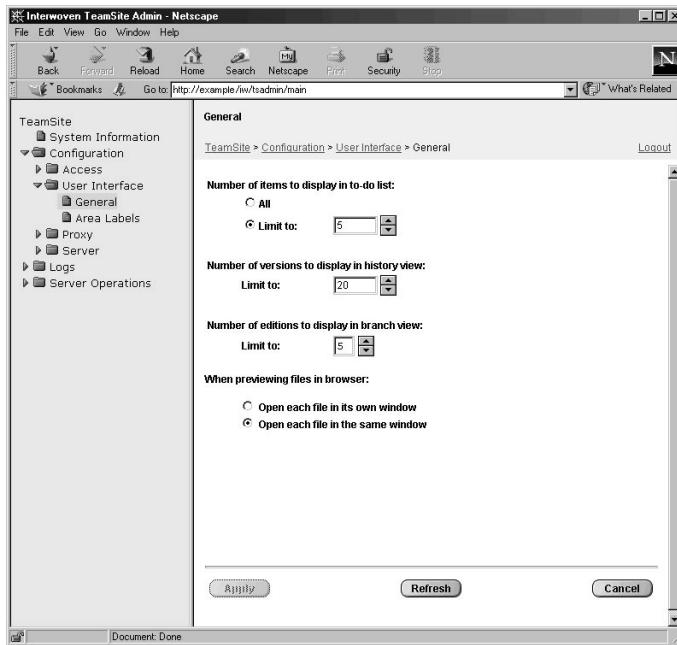
- **When submitting locked files**—Specifies whether anyone, or only the owner or creator of the lock, can submit locked files.
To configure this option manually in `iw.cfg`, see page 154.
- **When uploading files**—Specifies whether you want files uploaded only if TeamSite can establish a lock.
To configure this option manually in `iw.cfg`, see page 143.

Configuring General TeamSite GUI Preferences

You can configure a number of TeamSite GUI elements:

- Number of tasks to display in a To Do or Task list.
- Number of versions to display in the history view.
- Number of editions to display in the branch view.
- How to display files when previewing.

To set general aspects of the TeamSite GUI, navigate to **TeamSite > Configuration > User Interface > General**.



General preferences window, displaying the default settings

- **Number of items to display in to-do list**—Specifies how many jobs to display in an end-user's Task List.
To configure this option manually in `iw.cfg`, see page 143.
- **Number of versions to display in history view**—Limits the number of versions that display in the History view.
To configure this option manually in `iw.cfg`, see page 127.
- **Number of editions to display in branch view**—Limits the number of branches that display in the branch view.
To configure this option manually in `iw.cfg`, see page 126.
- **When previewing files in browser**—Specifies whether previewed files open in their own browser window, or in the same browser window.
To configure this option manually in `iw.cfg`, see page 133.

Changing Area Labels in the TeamSite GUI

You can change the labels that appear in the branch view of the TeamSite GUI, but you should use these options with extreme caution. TeamSite documentation and Interwoven Knowledge Base use the terms “branch,” “workarea,” “staging area,” and “edition” extensively. Changing these labels may confuse users.

To change area labels, navigate to **TeamSite > Configuration > User Interface > Area Labels**.

- **Branch**—Specifies the label for the section of the branch view that lists sub-branches of the main branch.
- **Staging**—Specifies the label for the section of the branch view that displays the staging area.
- **Edition**—Specifies the label for the section of the branch view that lists editions.
- **Work Area**—Specifies the label for the section of the branch view that lists workareas.

To configure these items manually in `iw.cfg`, see page 124.

Configuring the General Proxy Settings

In the General window, you can specify the host name and port number for the following servers:

- **Web Daemon**—Enables secure remote access to TeamSite.

For details about the TeamSite Web daemon, or for instructions on how to configure it manually, see “Configuring the TeamSite Web Daemon and Proxy Server” on page 175.

- **Proxy Server**—Enables virtualization of the Web sites.

For details about the TeamSite proxy server, or for instructions on how to configure it manually, see “Configuring the TeamSite Web Daemon and Proxy Server” on page 175.

- **Content Web server**—Serves the content of the Web sites.

For details about Web servers, or for instructions on how to configure them manually, see “Configuring Web Servers” on page 51.

- **Servlet Engine**—Serves the Java based portions of the TeamSite GUI.

For details about the TeamSite servlet engine, or for instructions on how to configure it manually, see “Servlet Engine” on page 152.

Note: Changes made through the General window affect only the corresponding sections in `iw.cfg`.

To change a setting:

1. Navigate to **TeamSite > Configuration > Proxy > General**.
2. In the **Host** field, enter the name of the server (for example, `example.com`).
3. In the **Port** field, enter the port number of the server.
4. Click **Apply**.

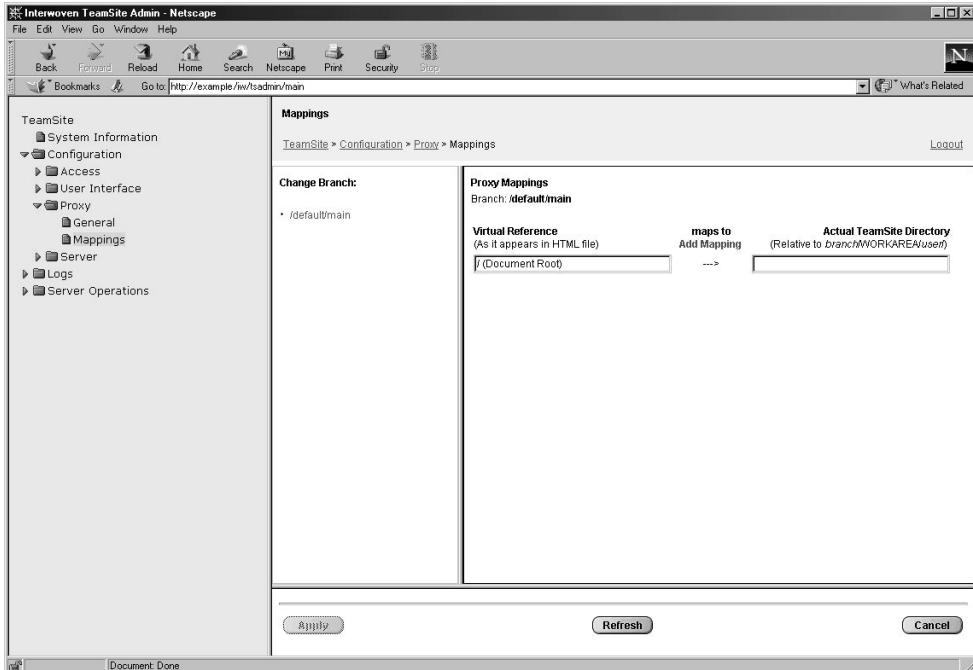
Configuring Proxy Mappings

In the Mappings window, you can specify the document root on a branch-by-branch basis, and can create specific directory mappings.

To change the document root of a branch:

1. Navigate to **TeamSite > Configuration > Proxy > Mappings**.

Configuring Proxy Mappings



Mappings window

2. In left pane, select a branch.
3. In right pane, in the **Actual TeamSite Directory** field next to `/ (DocumentRoot)`, enter the new document root.

To map a directory to a specific directory (other than the new document root):

1. Click **Add Mapping**.
2. Enter the reference as it would appear in the HTML file under **Virtual Reference**.
3. Enter the location it maps to, relative to the top of the user's workarea, under **Actual Teamsite directory**.
4. Click **Apply**.

To remove a specific mapping, click the **Remove** button for that mapping.

For details on the proxy server, or to configure mappings manually, see the section “Configuring the TeamSite Web Daemon and Proxy Server” on page 175.

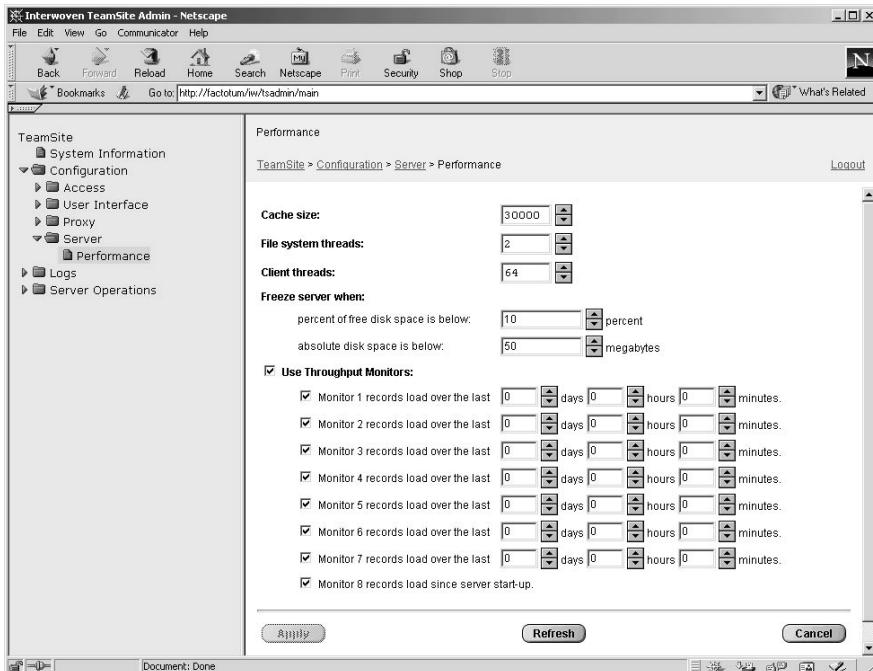
Configuring Server Performance

In the Performance window, you can specify:

- Cache size
- Number of file system threads
- Number of client threads
- Conditions under which you want to freeze the server
- How long you want various throughput monitors to record load

Changes to the cache size, the number of file system and client threads, and throughput monitors take effect only after TeamSite is restarted.

To configure server performance, navigate to **TeamSite > Configuration > Server > Performance**.



Performance window, displaying the default settings

- **Cache size**—Specifies the size of the TeamSite cache. The initial cache size setting should be approximately three times the number of files and directories on the largest branch. For example, if the largest branch contains 15,000 files and directories, you should set cache size to 45000. Maximum cache size is 400,000 entries. If your system's RAM is large enough (2 gigabytes or more), setting cache size to 200,000 can substantially improve performance for operations such as submit.

Caution: Data corruption can occur if the cache size is set to more than the memory can accommodate. For more information about cache size, and configuring cache size manually, see page 164.

- **File system threads**—Specifies the number of file system threads. The value should be set to approximately the number of CPUs on the TeamSite server.

Note: To configure file system threads manually, see page 165.

- **Client threads**—Specifies the number of simultaneous requests TeamSite can handle through the TeamSite GUI or command-line tools. These requests are very short-lived, so that threads are quickly freed for other users. If all threads are in use, TeamSite serializes requests. The default value for this setting is 64, and it should not be altered.

Note: To configure client threads manually, see page 165.

- **Freeze server when**—Specifies the conditions under which you want to freeze the TeamSite backing store.

For details about freezing the backing store, or to configure this feature manually, see page 167.

- **Throughput Monitors**—Specifies the monitors you want to activate and their monitoring intervals. For each of the monitors 1-7, you can specify a time interval in days, hours, or minutes, when you want that monitor to record the load on the server.

To configure throughput monitors manually, see page 166.

Configuring TeamSite Log Files

In the log files Settings window, you can specify:

- The type of information recorded in the Server, Event, and Trace log files.
- The number of events recorded in the Server, Event, and Trace log files.

The following table describes the contents of each log file:

Log	Contents
Server	Records the state of TeamSite over time. Tracks when the TeamSite server is started, shut down, mounted, and so on.
Event	Records activities on TeamSite. Tracks when files are submitted, published, branches created, and so on, including DiskLow, Freeze, ShutDown, StartUp, and Thaw events.
Trace	Records irregularities on the TeamSite server. Used by Interwoven Client Services to diagnose system performance and other issues.

To configure log files, navigate to **TeamSite > Logs > Settings**.

- **Number of events logged in update or submit**—Specifies the number of Submit and Get Latest operations to log for workareas. For example, if the value is set to 60, the Event log will contain the 60 most recent Submit or Get Latest operations (as opposed to the 60 most recent files that were submitted or updated).
- **Record individual file details during submit**—Lists, in the submit entries of the Event log, all the files in new or deleted directories.
- **Record individual file details during update**—Lists, in the submit entries of the Event log, all the files in new or deleted directories.

Viewing TeamSite Log Files

You can view the Server, Event, and Trace log files using the GUI. The following table describes the contents of each of these log files. For convenience, the default location of each log file is also listed.

Log	Location	Contents
Server	/var/adm/iwserver.log	Records the state of TeamSite over time. Tracks when the TeamSite server is started, stopped, mounted, and so on.
Event	/var/adm/iwevents.log	Records activities on TeamSite. Tracks when files are submitted, published, branches created, and so on, including DiskLow, Freeze, ShutDown, StartUp and Thaw events.
Trace	/var/adm/iwtrace.log	Record of any irregularities on the TeamSite server. Used by Interwoven Client Services to diagnose system performance issues.

To view a TeamSite log file:

1. Navigate to **TeamSite > Logs > Viewer** and select the log you want to view.

The following is an example of the TeamSite server log:



The screenshot shows a Netscape browser window titled "Interwoven TeamSite Admin - Netscape". The URL in the address bar is "http://example/iwtsadmin/main". The left sidebar contains a navigation tree with categories like TeamSite, Configuration, Logs, and Server Operations, with sub-options such as System Information, Access, User Interface, Proxy, Server, Settings, and Viewer (which includes Server Log, Event Log, Trace Log). The main content area is titled "Server Log" and displays the "TeamSite Server Log". The log file content is as follows:

```
iwstart: Spawning iwserver.2.6 (trace file: /var/adm/iwtrace.log)...
iwstart: Waiting up to 60 seconds for iwserver to come up... READY
iwatchboot: Mounting /iwmnt...
iwatchboot: Mounting /iwmnt...
iwatchboot: Mounted /iwmnt and /.iwmnt.
iwatchboot: Running iwstart /local/iw-store in /var/box/build/iw-home
iwatchboot: Running /var/box/build/iw-home/bin/iwfailsafe
ps: no controlling terminal
iwatchstart: The events log file is /var/adm/iwevents.log
iwatchstart: looking for the service pointer in the kernel ... found 601724c0
iwatchstart: reloading kernel module ... done
iwatchstart: patching kernel service pointer ... done
iwatchstart: checking for standard services
iwatchstart: nfsd running
iwatchstart: mountd running
iwatchstart: Spawning iwserver.2.6 (trace file: /var/adm/iwtrace.log)...
iwatchstart: Waiting up to 60 seconds for iwserver to come up... READY
```

Below the log content, there is a "Page size:" dropdown set to 24 lines, and a set of navigation buttons: <<, <, All, >, >>.

Log Viewer, displaying the Server log

2. Set the page size by entering the number of lines you want to view at any given time or click **All** if you want the entire contents of the log displayed.

Note: TeamSite servers that have been running for a long time might have extremely large log files. Thus, if you click **All** to view the entries for such a log file, it may take several moments for all the entries to load.

3. Navigate through the log file by clicking <<, <, >, or >>.
 - Click > to move one page toward the most recent entry (the end of the log file).
 - Click < to move one page toward the oldest entry (the beginning of the log file).
 - Click >> to move to the last page of the log file (the most recent entries).
 - Click << to move to the first page of the log file (the oldest entries).

Performing Server Operations

The Server Operations window enables you to perform the following operations:

- Abort
- Freeze and Unfreeze
- Reset

The procedures associated with these operations are described in the following sections.

Abort

To abort a server operation:

1. Navigate to **TeamSite > Server Operations > Abort**.

Currently active operations display in the **Select an operation to abort** field.

2. Select the operation you want to abort.
3. Click **Apply**.

The operation is terminated at the earliest possible time.

4. Click **Refresh** to refresh the list of currently active operations.

Freeze or Unfreeze

To freeze the TeamSite server:

1. Navigate to **TeamSite > Server Operations > Freeze/Unfreeze**.
2. Select **Freeze**.
3. Enter the number of seconds you want the TeamSite server to be frozen.
4. Select if you only want the freeze to operate on batch jobs.
5. Click **Apply**.

To unfreeze the TeamSite server, click the **Unfreeze** option, then click **Apply**.

The freeze and unfreeze operations do not enable you to specify a backing store—they operate on all stores (freezes or unfreezes all of them at a time). If you want to specify a specific store in a MultiStore environment, use the `iwfreeze` CLT. For more information about the `iwfreeze` command-line tool, consult the *TeamSite Command-Line Tool Reference*.

Reset

This operation tells the server to reread its configuration files (equivalent to the `iwreset` command). For more information on the `iwreset` command-line tool, consult the *TeamSite Command-Line Tool Reference*.

To reset the TeamSite server:

1. Navigate to **TeamSite > Server Operations > Reset**.
2. Click **Reset Server**.

Chapter 5

Configuring the TeamSite Server

Most of the settings for the TeamSite server are configured in the main configuration file, `/etc/iw.cfg` (default location).

Some settings are configured in the following files:

- `iw-home/local/config/submit.cfg`
- `iw-home/local/config/autoprivate.cfg`
- `iw-home/local/config/iwtemplates.cfg`
- `iw-home/local/config/file_encoding.cfg`

Changes to most of these configuration options take effect within a few minutes (although for options that affect the TeamSite GUI, you may have to clear your browser cache in order to see the changes). For these options to take immediate effect, use the `iwreset` command-line tool (CLT). Configuration options that require TeamSite to be restarted in order to take effect are marked throughout this chapter.

For workflow-related configuration options, see the *TeamSite Workflow Developer's Guide*.

For TeamSite Templating configuration, see the *TeamSite Templating Developer's Guide*.

Option	Configuration file	Page
Configuring GUI appearance		
Configuring TeamSite area labels	<code>iw.cfg</code>	page 124
Configuring the number of displayed editions	<code>iw.cfg</code>	page 126
Configuring the number of displayed versions	<code>iw.cfg</code>	page 127
Individual user home page settings	Entity database	page 127
Configuring GUI functionality		
Enabling/disabling Editor publish capability	<code>iw.cfg</code>	page 128
Selectively enabling or disabling SmartContext Editing	<code>iw.cfg</code>	page 128

Option	Configuration file	Page
Adding edit and assign task links to Web pages (Casual Contributor interface)	<i>iw.cfg</i>	page 129
Setting the default LaunchPad interface	<i>iw.cfg</i>	page 131
Setting unique server names for LaunchPad to recognize	<i>iw.cfg</i>	page 132
Setting the login authentication expiration	<i>iw.cfg</i>	page 132
Setting the number of GUI preview windows	<i>iw.cfg</i>	page 133
Adding custom menu items	<i>iw.cfg</i>	page 134
Configuring submit button behavior	<i>iw.cfg</i>	page 139
Disabling menu items	<i>iw.cfg</i>	page 140
Disabling directory operations	<i>iw.cfg</i>	page 142
Disabling unlocked file auto-upload	<i>iw.cfg</i>	page 143
Setting the number of jobs listed in the To Do List	<i>iw.cfg</i>	page 143
Configuring job attributes and filters	<i>iw.cfg</i>	page 144
Configuring email settings	<i>iw.cfg</i>	page 145
Configuring server functionality		
Setting the encoding of <i>iw.cfg</i>	<i>iw.cfg</i>	page 146
Setting authentication type	<i>iw.cfg</i>	page 146
Setting the webserver UID	<i>iw.cfg</i>	page 152
Configuring the Web daemon	<i>iw.cfg</i>	page 152
Configuring the servlet engine	<i>iw.cfg</i>	page 152
Setting the main branch locking model, owner and group	<i>iw.cfg</i>	page 153
Configuring submit capabilities on locked files	<i>iw.cfg</i>	page 154
Configuring the events logged in the submit and update logs	<i>iw.cfg</i>	page 154
Setting branch and workarea security	<i>iw.cfg</i>	page 154
Setting default permissions	<i>iw.cfg</i>	page 155
Configuring group remapping	<i>iw.cfg</i>	page 156
Setting TeamSite file locations	<i>iw.cfg</i>	page 156
Configuring Autoprivacy	<i>autoprivate.cfg</i>	page 158

Option	Configuration file	Page
Configuring New File templates	<code>iwtemplates.cfg</code>	page 161
Configuring use of the proxy server	<code>iw.cfg</code>	page 163
Configuring the TeamSite server locale	<code>iw.cfg</code>	page 163
Configuring encoding rules for text files	<code>file_encoding.cfg</code>	page 283
Configuring server performance		
Setting cache size	<code>iw.cfg</code>	page 164
Setting RPC thread count	<code>iw.cfg</code>	page 165
Setting file system threadcount	<code>iw.cfg</code>	page 165
Setting file system active area cache	<code>iw.cfg</code>	page 166
Configuring throughput monitors	<code>iw.cfg</code>	page 166
Detecting low disk space and inodes	<code>iw.cfg</code>	page 167
Configuring submit filtering		page 167
Changing file attributes at submit time	<code>submit.cfg</code>	page 167
RCS macro expansion	<code>submit.cfg</code>	page 172
Configuring the TeamSite proxy server		page 175
Configuring proxy server operation	<code>iw.cfg</code>	page 177
Resolving relative and absolute URLs	<code>iw.cfg</code>	page 178
Resolving fully-qualified URLs	<code>iw.cfg</code>	page 182
Redirecting TeamSite views to different areas	<code>iw.cfg</code>	page 186
Configuring TeamSite to use different webservers	<code>iw.cfg</code>	page 189
Configuring external remappings	<code>iw.cfg</code>	page 190
Host header remappings	<code>iw.cfg</code>	page 191
Configuring SSI remappings	<code>iw.cfg</code>	page 192
Configuring proxy failover	<code>iw.cfg</code>	page 192
Configuring TeamSite Embedded Failsafe		page 195
Disabling Embedded Failsafe	<code>iw.cfg</code>	page 195

Configuring GUI Appearance

Configuring TeamSite Area Labels

You can change the labels that appear in the TeamSite GUI (WebDesk Pro) branch view by editing the area label lines in `iw.cfg`. Use this option with caution, however, because the “branch,” “staging area,” “edition,” and “workarea” terms are used throughout TeamSite documentation and in the Interwoven Knowledge Base, and changing these labels may cause confusion among users.

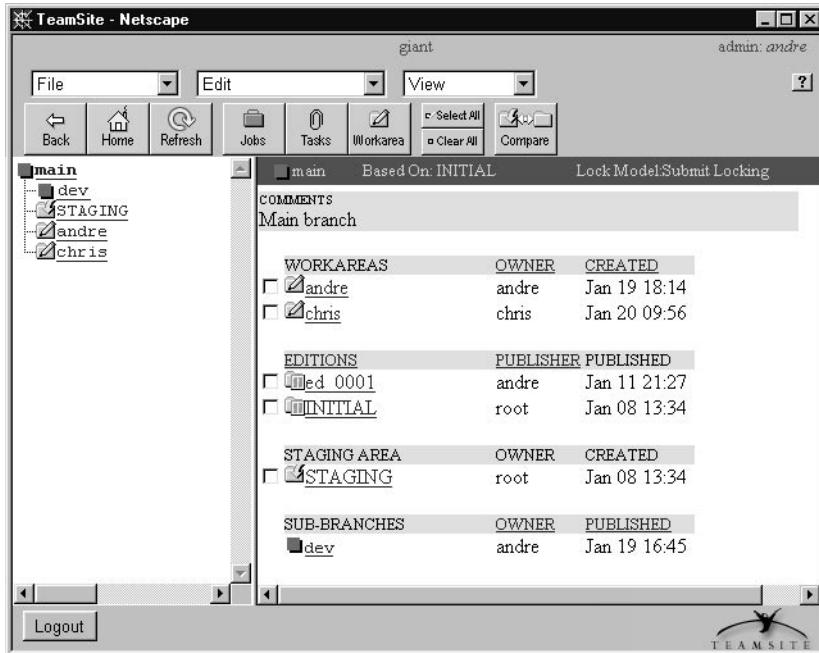
To change these labels, edit the following lines in the `[iwcfg]` section of `iw.cfg`. If these lines do not appear in `iw.cfg`, add them as shown below:

```
branch_label=new_branch_label  
staging_label=new_staging_area_label  
edition_label=new_edition_label  
workarea_label=new_workarea_label
```

For example, with the default values of:

```
branch_label=SUB-BRANCHES  
staging_label=STAGING AREA  
edition_label=EDITIONS  
workarea_label=WORKAREAS
```

The branch view looks like:

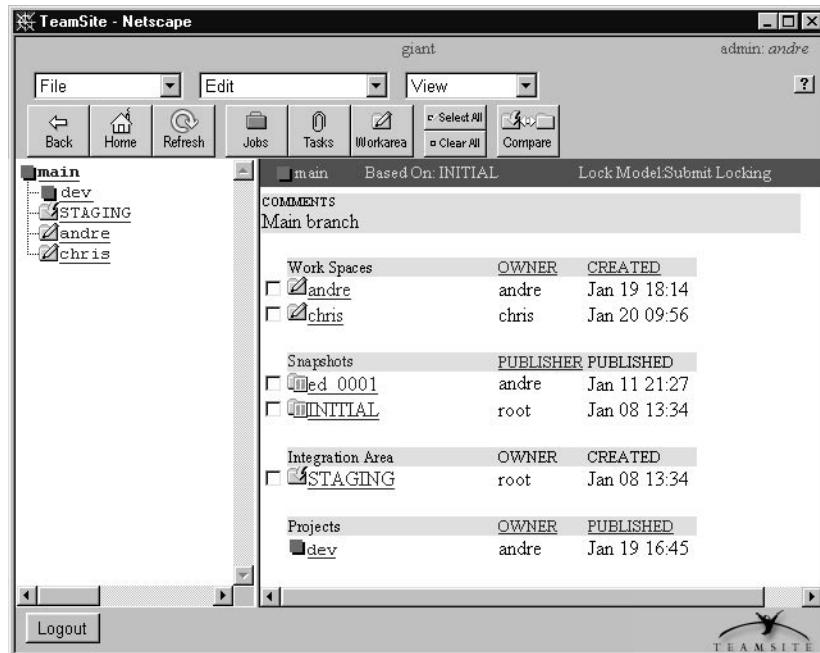


Default TeamSite labels

However, giving these labels other values, such as:

```
branch_label=Projects  
staging_label=Integration Area  
edition_label=Snapshots  
workarea_label=Work Spaces
```

Would change the labels in the branch view to:



The TeamSite Area Labels feature allows you to change the labels that appear in the Work Window in the branch view

Configuring Edition Views

You can configure the number of editions you want to see in the branch view of the GUI. To view prior editions, click the **Show all editions** link in the GUI.

To set the number of editions to display, edit the edition list line in the [iwcgi] section of `iw.cfg`, as shown below:

```
edition_list_limit=number_of_editions
```

For example:

```
edition_list_limit=10
```

would configure TeamSite to display only the ten most recent editions.

If this line does not appear in `iw.cfg`, add it as shown above. The default value is 5 (by default, TeamSite will display the five most recent editions in the branch view).

To show all editions by default, comment out the `edition_list_limit` line by adding a # to the beginning of the line.

Configuring History Views

You can configure the number of versions shown in the History view of the TeamSite GUI. To configure this option, use the `view_history_limit` parameter in the `[iwcgi]` section of `iw.cfg`. For example:

```
view_history_limit=5
```

would restrict the History view to show only the five most recent versions of a file. All versions would still exist; however, only five would be displayed.

User Profiles

The SetHomePage functionality (where users can set their Home page) in the WebDesk Pro GUI now stores the homepage information in the entity database instead of the `iwprofiles` directory.

If you are upgrading to TeamSite 5.5.1, you must run the `iwprefconv` CLT (as described in the *Command-Line Tools* manual) once to copy any existing homepage information from the `iwprofiles` directory (`iw-home/local/iwprofiles`) to the entity database (`iw-home/local/entities/data`).

Do not modify these files manually. These files are automatically generated by the TeamSite server and updated as needed.

Configuring GUI Functionality

Disabling Editor Publish Capability

TeamSite allows you to turn off Editors' ability to publish. You cannot turn this option off for selected Editors; it applies to all Editors on all branches.

To turn off the Publish capability for Editors:

If applicable, remove the comment mark (#) from the `editor_publish` line in the [main] section of `iw.cfg`. If `iw.cfg` does not contain this line, add it as shown below.

```
editor_publish=no
```

Enabling and Disabling SmartContext Editing

You can selectively enable or disable SmartContext Editing for different workareas or files by adding lines to the [iwproxy_smartcontextedit_allowed] section of `iw.cfg`. If this section does not exist, SmartContext Editing is enabled by default.

The [iwproxy_smartcontextedit_allowed] section contains one `_default` line, which specifies whether SmartContext Editing is turned on or off in any area or for any file not otherwise specified. This section can also contain any number of `_regex` lines. Each `_regex` line uses a case-insensitive regular expression to specify areas or files, and then specifies whether SmartContext Editing is enabled or disabled for the specified items. A `_regex` line has the following case-insensitive syntax:

```
_regex=regular-expression=yes|no
```

`_regex` lines are order-dependent. For example, the following [iwproxy_smartcontextedit_allowed] section turns SmartContext Editing on by default, and it explicitly turns it on for all files in all of Andre's workareas on all branches. It then turns SmartContext Editing off for all CGI files. Because the line turning SmartContext Editing on for Andre's workareas comes first, he will be able to use SmartContext Editing for CGI files in his workarea:

```
[iwproxy_smartcontextedit_allowed]
_default=yes
_regex=(.*)/WORKAREA/andre/.*=yes
_regex=\.cgi(\?.*)?=$=no
```

The Casual Contributor Interface: Adding Editing and Task Links to Web Pages

You can give Authors the ability to access WebDesk file editing and task features directly from any Web page or email message by adding one or more URL links to the source file or message. When a user clicks on one of these links, the user will be taken directly to the appropriate functionality within TeamSite.

If the user is not already authenticated, the TeamSite login screen is displayed.

To create a link, use the following syntax:

http://servername/iw/webdesk/function?vpath=filename

or

http://servername/iw/webdesk/taskaction?taskid=taskid

where the variables are defined as follows:

<i>servername</i>	The name of the server
<i>function</i>	One of: <ul style="list-style-type: none"> • assign—prompts the user to create a new task using <code>default_assign.wft</code> with the specified file attached. • edit—opens the file for editing in WebDesk. • sce—opens the specified file in a browser for use with SmartContext Editing. • tag—opens the Metadata Capture dialog. If MetaTagger 3.0 is installed, this will launch its Metadata Capture dialog. • details—displays the File Properties window. • visualdiff—displays the Visual Difference window, comparing the specified file with the version in the staging area.
<i>filename</i>	Directory path and filename of the file.
<i>taskaction</i>	One of: <ul style="list-style-type: none"> • task—displays the corresponding task ID. • transitiontask—opens the Task Transition dialog. • taketask—grants the user ownership of the specified group task (<i>taskid</i> must refer to a group task).
<i>taskid</i>	Integer ID of a task.

For example, if you want to create a link from a Web page that opens the source file `sample.html` in Edit mode, from the server `example`, and the source file is in `/default/main/example/sample.html`, you would enter the following URL in the link:

`http://webdev/iw/example/edit?vpath=/default/main/dev/sample.html`

Sample HTML including the above link might look something like this:

```
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=UTF-8">
<TITLE>TeamSite URL Example</TITLE>
</HEAD>

<BODY>
<H1>TeamSite URL Example</H1>
<P>
With the TeamSite WebDesk user interface, it is easy for users to
<A HREF="http://example/iw/webdesk/edit?vpath=/default/main/dev/
sample.html">edit</A> any file in your asset base!
</P>
</BODY>
</HTML>
```

You can also embed a task link within an email message or Web page by using the following syntax:

`http://servername/iw/webdesk/task?taskid=number`

where *servername* is the name of the server and *number* is the task ID.

Setting the Default LaunchPad Interface

TeamSite offers both an applet and a standalone application version of LaunchPad. The entry in the [iwcgi] section of *iw.cfg* is set to use the applet by default:

```
[iwcgi]
use_launchpad_applet=true
```

If you prefer for all clients to use the LaunchPad application, change the `use_launchpad_applet` setting to `false`.

Note that clients who use Netscape on the Macintosh or who have Java turned off in their browsers cannot use the applet. They will always use the application.

Setting Unique Server Names for LaunchPad to Recognize

To enable LaunchPad to differentiate between different TeamSite servers, set a unique server name in `iw.cfg`:

```
launchpad_hostname=hostname
```

where *hostname* is the server name.

Setting Login Authentication Expiration

If users log in to WebDesk, then close the window, then reopen WebDesk again within the same user session on a machine within a 24-hour period, TeamSite recognizes them from their previous login and does not require them to re-enter their username and password. This is important for users of the Casual Contributor interface, who will be transferred directly to the relevant part of WebDesk if they have previously logged in within the authentication period.

The default login authentication expiration is 24 hours. To change this value, edit the `ui_login_lifetime` setting in the `[authentication]` section of `iw.cfg` using the following syntax:

- `ui_login_lifetime=ahbmcs` (where *a* represents the number of hours, *b* represents the number of minutes, and *c* represents the number of seconds)
- `ui_login_lifetime=xYyMzD` (where *x* represents the number of years, *y* represents the number of months, and *z* represents the number of days)

When specifying years, months, or days, Y, M, or D must be upper-case.

For example, to set the login authentication expiration to 2 days, enter `2D`, as follows:

```
[authentication]  
ui_login_lifetime=2D
```

To set the login authentication expiration to 8 hours, 15 minutes, and 7 seconds, enter `8h15m7s`, as follows:

```
[authentication]  
ui_login_lifetime=8h15m7s
```

To set the login authentication expiration to 1.5 years, enter `1Y6M`, as follows:

```
[authentication]
ui_login_lifetime=1y6m
```

If you want the login authentication to never expire, enter `infinite`, as follows:

```
[authentication]
ui_login_lifetime=infinite
```

Similarly, the `cookie_lifetime` setting in the `iw.cfg` file controls the term for which browser cookies are issued when users log in to WebDesk or WebDesk Pro. A value of 0 (zero) creates session cookies, such that users are automatically logged out by exiting the browser. A value of infinite (which is the default) causes persistent cookies to be issued, such that users stay logged in until the `ui_login_lifetime` is reached or the user deliberately logs out. No other values should be used.

Configuring Preview Windows

When you click on the name of a file in the TeamSite GUI, TeamSite launches a browser window so you can preview it. By default, a new browser window is launched each time you click on a file name. However, you can configure TeamSite to reuse the same window each time you preview a file.

To configure the number of browser windows TeamSite launches, you must edit the `single_browser_window` line in the `[iwcgi]` section of `iw.cfg`. If this line does not exist, add it as shown below.

To reuse the same window each time you preview a file, set `single_browser_window=TRUE`. To launch a new browser window each time you preview a file, set `single_browser_window=FALSE` or comment the line out altogether. This setting will apply to all users on all branches of the TeamSite server.

```
[iwcgi]
single_browser_window=TRUE
```

Custom Menu Items

You can add custom menu items to either WebDesk or WebDesk Pro. These menu items can call either CGI scripts or HTML pages.

Note that TeamSite includes all custom **File** menu items in the **File Options** drop-down list for each file listed in the Task Details screen of WebDesk Pro. TeamSite checks `iw.cfg` for **File** custom menu items, and adds them to the **File Options** drop-down list. Custom menu items for other menus (**Edit** and **View**) are not included in the Task Details screen in WebDesk Pro.

About CGI Scripts

CGI scripts that are added to the TeamSite interface are executed via the TeamSite CGI launcher/wrapper, which calls the CGI program and sets the environment variables that would be set by the webserver.

Creating Custom CGI Scripts

The CGI wrapper makes certain variable=value pairs available, depending on the user's current location and any items that are selected.

The user's username and role are always available. Also available is the vpath of the current location, the mount path of the TeamSite mount point, and the directory paths, object ids, and names of the current branch and archive, and (if applicable) of the current sub-branch, area, and directory. The `page_type` variable indicates what type of TeamSite area the user is currently in, and the `subpage_type` variable (available when the user is not on a branch page), indicates whether the user is currently in a sub-directory or the root directory of his current area. Each item selected has four variables associated with it: type, object id, name, and path.

For example, user `andre` logged in as Master might navigate into the `htdocs` directory in his workarea on the `main` branch, and select the checkboxes next to two directories. The following variable=value pairs would then be available:

```

iw_prog_name=custom_script.cgi
wrapper_version=1
vpath=/default/main/WORKAREA/andre/htdocs
mount_path=/iwmnt
directory_id=0x0000007b00000079000000b9
directory_name=htdocs
directory_path=/iwmnt/default/main/ WORKAREA/
andre/htdocs

area_id=0x000021000000000000000007b
area_name=andre
area_path=/iwmnt/default/main/WORKAREA/andre

branch_id=0x000022500000000000000006d
branch_name=main
branch_path=/iwmnt/default/main

archive_id=0x000020200000000000000001
archive_name=default
archive_path=/iwmnt/default

subpage_type=sub_directory
page_type=workarea
session=AAAAAQAFYW5kcmUAAAAMjAwMS5hbWRYZTWZhj4A
page_id=8
user_name=andre
user_role=master

type_0=directory
objid_0=0x0000007b000000b9000000e9
name_0=corporate
path_0=/iwmnt/default/main/WORKAREA/andre/
htdocs/corporate

type_1=directory
objid_1=0x0000007b000000b9000000e5
name_1=news
path_1=/iwmnt/default/main/WORKAREA/andre/
htdocs/news

```

— Name of the CGI script
 — Version of the wrapper
 — Vpath of current location
 — TeamSite mount point

} Attributes of current directory

} Attributes of current area

} Attributes of current branch

} Attributes of current archive

— Type of directory
 — Type of area
 — For impersonation use
 — For internal use

} User's name and role

} Type, object id, name, and directory path for the first item selected

} Type, object id, name, and directory path for the second item selected

Adding Custom CGI Scripts to WebDesk and WebDesk Pro

To add a custom CGI script to the TeamSite GUI (WebDesk or WebDesk Pro):

1. Create a CGI program in `iw-home/httpd/iw-bin`.

2. Add the following line to the `[iwcgi]` section of `iw.cfg`:

```
custom_menu_item_identifier="MenuName", "MenuItemName",
"CGIProgramName", "RolesList", "WindowAttributes",
"WindowName", "500"
```

where `identifier` is a unique identifier for the menu item, and the parameters are as follows. Note that some parameters differ depending on whether you want the menu item to appear in WebDesk or WebDesk Pro:

Parameter	Description	WebDesk	WebDesk Pro
<code>MenuName</code>	The menu to add the entry to.	Required Possible values are <code>Edit</code> or <code>View</code> .	Required Possible values are <code>File</code> , <code>Edit</code> , or <code>View</code> .
<code>MenuItemName</code>	The name of the menu item as it appears to the user.	Required	Required
<code>CGIProgramName</code>	The CGI program to execute (must be in <code>iw-home/httpd/iw-bin</code>). The program name may not contain spaces.	Required	Required
<code>RolesList</code>	The comma-separated list of roles who will have access to this menu item.	Required Must contain <code>author</code> or <code>all</code> .	Optional ¹ (required if <code>WindowAttributes</code> is specified) Can contain <code>author</code> , <code>editor</code> , <code>admin</code> , <code>master</code> , or <code>all</code> .

Parameter	Description	WebDesk	WebDesk Pro
<i>WindowAttributes</i>	Specifies the characteristics of the window.	Required See table below for possible attributes.	Optional ² (required if <i>WindowName</i> is specified) See table below for possible attributes.
<i>WindowName</i>	Specifies the name of the window. If the menu item does not need a window, specify _nowindow.	Required	Optional
500	Specifies that this menu item will appear in WebDesk. Menu items that appear in WebDesk will also appear in WebDesk Pro.	Required	Omit if you want the menu item to appear in WebDesk Pro but not in WebDesk. If included, follow WebDesk requirements.

1. If *RolesList* is not specified, all roles are assumed.
2. If *WindowAttributes* is not specified, the defaults are:
`resizable=yes,scrollbars=no,menubar=yes,width=640,height=480`

WindowAttributes are specified as follows. Unless otherwise specified, all window attributes are of the form *value=yes|no*.

<i>toolbar</i>	Specifies whether or not the browser toolbar will appear.
<i>location</i>	Specifies whether the Location input field (for entering URLs) will appear.
<i>directories</i>	Specifies whether directory buttons will appear.
<i>status</i>	Specifies whether the status line will appear.
<i>scrollbars</i>	Enables scrollbars.
<i>resizable</i>	Allows the user to resize the window.
<i>menubar</i>	Specifies whether the browser menu bar will appear.
<i>width</i>	Specifies the width of the window (in pixels).
<i>height</i>	Specifies the height of the window (in pixels).

For example:

```
custom_menu_item_show_env="File", "Environment", "show_env.cgi",
"admin, master", "width=640,height=450,scrollbars=yes,resizable=yes"
```

creates a new custom menu item called `show_env`. This menu item will appear in the `File` menu of WebDesk Pro, and it will be called **Environment**. It will call the CGI program `show_env.cgi`, and the menu item will be available only to Administrators and Master users. The window that appears will be 640 pixels wide by 450 pixels high, it will have scrollbars, and it will be resizable.

```
custom_menu_item_reports="View", "Reports", "report.cgi", "admin,
master, author", "scrollbars=yes,resizable=yes,width=640,height=545",
"reports", "500"
```

creates a new custom menu item called `reports`. This menu item will appear in the `View` menu of both WebDesk and WebDesk Pro, and it will be called **Reports**. It will call the CGI program `report.cgi`, and the menu item will be available only to Authors, Administrators and Master users. The window that appears will be 640 pixels wide by 545 pixels high, it will have scrollbars, and it will be resizable. The window will be named `reports`.

3. Log in and select the menu where you added the new item. You will see the new menu item at the bottom of the menu. When you select this item, a separate window will display the output of your CGI program.

Adding HTML Pages to WebDesk and WebDesk Pro

You can also launch custom HTML files in a separate window from TeamSite. The HTML file will be called directly from the web browser, without any special preprocessing.

To add an HTML file to TeamSite (WebDesk or WebDesk Pro):

1. Create the HTML file.
 2. Add the following line to the [iwcfgi] section of `iw.cfg`:
- ```
custom_menu_item_identifier="MenuName", "MenuItemName", "file:URL",
"RolesList", "WindowAttributes", "WindowName", "500"
```

where all parameters except `file:URL` are specified as described in “Adding Custom CGI Scripts to WebDesk and WebDesk Pro” on page 136. `URL` is the URL of the HTML file to call. For example, an entry that calls the file `www.example.com/internal/localhelp.html` might look like:

```
custom_menu_item_help="View", "Local help", "file:www.example.com/internal/localhelp.html"
```

`WindowName` and `500` are required for adding custom HTML pages to WebDesk. If you only want a page to appear in WebDesk Pro, omit the `500` parameter. In this case, `WindowName` is optional.

## Configuring Submit Button Behavior

The **Submit** button can be configured to either submit files directly to the staging area (Submit-Direct), or to use the default Submit workflow process (Submit-Process). By default, the Submit button will use workflow for all roles.

To configure the Submit button behavior on a per-role basis, add the following section to `iw.cfg`:

```
[submit_button]
submit_direct=roleslist
submit=roleslist
```

where `roleslist` specifies the roles that use this option (`editor`, `admin`, `master`, or `all`—Authors must always use the Submit workflow process).

For example:

```
[submit_button]
submit_direct=admin, master
submit=editor
```

would configure the **Submit** button to submit files directly to the staging area for all Administrators and Master users, but to use the Submit workflow process for all Editors. Authors would still use the Submit workflow process.

If you disable the Submit button for any role (that is, if you include `submit=roleslist` in both the `[ui_remove_menu_items]` section and the `[ui_disable_directories]` section—see

below), then this section will have no effect for that role, as there will be no Submit button to configure.

## Disabling Menu Items

You can now disable TeamSite menu items and buttons on a per-role basis. To disable a TeamSite menu item, add a new section to TeamSite's main configuration file, `iw.cfg`, as follows:

```
[ui_remove_menu_items]
menuitemname="roleslist"
```

where `menuitemname` is the name of the menu item you want to disable, and `roleslist` is a comma-separated list of roles (for example `author`, `editor`). The menu item will be disabled in WebDesk Pro for all roles specified, and in WebDesk if `author` is specified. If the menu item has a corresponding button, it will be removed from the WebDesk Pro Button Bar for these roles (the **Compare Any** menu item and the **Compare with Staging** button are both governed by the `compare` value of `menuitemname`).

You can add multiple lines to a `[ui_remove_menu_items]` section. For example, the following `[ui_remove_menu_items]` section turns off the **Delete** and **Move** menu items for Editors and Authors.

```
[ui_remove_menu_items]
delete="editor,author"
move="editor,author"
```

This is a complete list of values of `menuitemname` for the `[ui_remove_menu_items]` section. Items marked with an asterisk (\*) only apply if TeamSite Templating is installed:

| Value   | Disabled Menu Item  | Disabled Button                          |
|---------|---------------------|------------------------------------------|
| assign  | File > Assign       | Assign                                   |
| compare | File > Compare Any  | Compare (compares with the staging area) |
| copy    | File > Copy         | N/A                                      |
| copyto  | File > Copy to Area | N/A                                      |
| delete  | File > Delete       | N/A                                      |

| <b>Value</b>    | <b>Disabled Menu Item</b>         | <b>Disabled Button</b>                    |
|-----------------|-----------------------------------|-------------------------------------------|
| *editdcr        | <b>Edit &gt; Edit Data Record</b> | N/A                                       |
| editfile        | <b>Edit &gt; Edit File</b>        | <b>Edit File</b>                          |
| editfilewith    | <b>Edit &gt; Edit File With</b>   | N/A                                       |
| file_properties | <b>File &gt; File Properties</b>  | N/A                                       |
| *genHTML        | <b>File &gt; Generate HTML</b>    | N/A                                       |
| getlatest       | <b>File &gt; Get Latest</b>       | <b>Get Latest</b>                         |
| history         | <b>View &gt; History</b>          | N/A                                       |
| import_files    | <b>File &gt; Import Files</b>     | N/A                                       |
| listlocks       | <b>View &gt; List Locks</b>       | N/A                                       |
| listmodified    | <b>View &gt; List Modified</b>    | N/A                                       |
| lock            | <b>Edit &gt; Lock</b>             | N/A                                       |
| move            | <b>File &gt; Move</b>             | N/A                                       |
| new_branch      | <b>File &gt; New Branch</b>       | N/A                                       |
| *newdcr         | <b>File &gt; New Data Record</b>  | N/A                                       |
| newdir          | <b>File &gt; New Directory</b>    | N/A                                       |
| newfile         | <b>File &gt; New File</b>         | N/A                                       |
| newJob          | <b>File &gt; New Job</b>          | N/A                                       |
| new_workarea    | <b>File &gt; New Workarea</b>     | N/A                                       |
| private         | <b>Edit &gt; Private</b>          | N/A                                       |
| public          | <b>Edit &gt; Public</b>           | N/A                                       |
| publish         | <b>File &gt; Publish</b>          | N/A                                       |
| *regenHTML      | <b>File &gt; Regenerate HTML</b>  | N/A                                       |
| rename          | <b>File &gt; Rename</b>           | N/A                                       |
| setHomePage     | <b>Edit &gt; Set Home Page</b>    | N/A                                       |
| submit          | <b>File &gt; Submit</b>           | <b>Submit</b> (configurable—see page 139) |
| submit_direct   | <b>File &gt; Submit-Direct</b>    | <b>Submit</b> (configurable—see page 139) |

| <b>Value</b> | <b>Disabled Menu Item</b>   | <b>Disabled Button</b> |
|--------------|-----------------------------|------------------------|
| submitlog    | <b>View &gt; Submit Log</b> | N/A                    |
| task_todo    | <b>View &gt; To Do List</b> | <b>To Do</b>           |
| unlock       | <b>Edit &gt; Unlock</b>     | N/A                    |
| updatelog    | <b>View &gt; Update Log</b> | N/A                    |
| viewfile     | <b>Edit &gt; View File</b>  | <b>View File</b>       |

## Disabling Directory Operations

You can now disable certain operations' ability to act on directories in WebDesk Pro, on a per-role basis. To disable a TeamSite menu item's abilities to act on directories, add a new section to the TeamSite main configuration file, `iw.cfg`, as follows:

```
[ui_disable_directories]
menuitemname="roleslist"
```

where `menuitemname` is the name of the menu item you want to disable for directories, and `roleslist` is a comma-separated list of roles (for example, `author`, `editor`). Specify `all` to disable the menu item for all roles. The menu item will no longer act on directories when it is invoked by a user who is logged in to WebDesk Pro with one of these roles.

You can only disable menu items' abilities to act on directories if they would ordinarily be able to do so.

You can add multiple lines to a `[ui_disable_directories]` section. For example, the following `[ui_disable_directories]` section disables Editors' and Authors' abilities to delete or move directories.

```
[ui_disable_directories]
delete="editor,author"
move="editor,author"
```

This is a complete list of values of *menuitemname* for the [ui\_disable\_directories] section:

| <b>Value</b>  | <b>Disabled Menu Item</b>      | <b>Disabled Button</b>                    |
|---------------|--------------------------------|-------------------------------------------|
| assign        | <b>File &gt; Assign</b>        | <b>Assign</b>                             |
| copy          | <b>File &gt; Copy</b>          | N/A                                       |
| copyto        | <b>File &gt; Copy to Area</b>  | N/A                                       |
| delete        | <b>File &gt; Delete</b>        | N/A                                       |
| getlatest     | <b>File &gt; Get Latest</b>    | <b>Get Latest</b>                         |
| move          | <b>File &gt; Move</b>          | N/A                                       |
| private       | <b>Edit &gt; Private</b>       | N/A                                       |
| public        | <b>Edit &gt; Public</b>        | N/A                                       |
| rename        | <b>File &gt; Rename</b>        | N/A                                       |
| submit        | <b>File &gt; Submit</b>        | <b>Submit</b> (configurable—see page 139) |
| submit_direct | <b>File &gt; Submit-Direct</b> | <b>Submit</b> (configurable—see page 139) |

## Disabling Unlocked File Auto-Upload

By default, TeamSite allows you to upload files even if it cannot establish a lock on them. To disable this feature so that files are uploaded only if TeamSite can establish a lock, add the following line to the [iaproxy] section of `iw.cfg`:

```
allow_unlocked_file_upload=no
```

To turn unlocked file uploading back on, either remove this line from `iw.cfg` or set it to `yes`.

## Setting the Number of Jobs Listed in the To Do List

The `iw.cfg` configuration file allows you to set the maximum number of jobs to be listed in the To Do List. To configure this option, add the following line to the [iworkflow\_ui] section of `iw.cfg`.

If this section does not exist, create it as follows:

```
[iw_workflow_ui]
max_job_count_per_page=number_of_jobs|all
```

The default is 55 jobs per page. You can change the number of jobs by specifying a value. If you specify all, all the jobs fulfilling a particular view display on a single page. In Windows 98, if you have over 100 jobs and specify all, the workflow screen may be exceedingly slow.

## Configuring Job Attribute Filters and Settings

Job attributes are configurable properties of individual jobs. These attributes show up in the workflow section of the TeamSite GUI. You can set these attributes through the GUI, and you can choose to display only the jobs that have certain attribute settings. The names of these attributes and their possible settings are configured in the [iw\_workflow\_ui] section of iw.cfg.

For example, if you have attributes called Category, Month, and Date, you can use drop-down menus in the GUI to view only jobs with specific values of Category, Month, and Date (for example, Category=Marketing, Month=November, Date=24).

You can configure up to three attributes. Each attribute can have any number of possible values. Values are separated by colons.

The [iw\_workflow\_ui] section of iw.cfg has the following format:

```
[iw_workflow_ui]
attribute1=attributename1
values1=valuelist1
attribute2=attributename2
values2=valuelist2
attribute3=attributename3
values3=valuelist2
```

where *attributename1*, *attributename2*, and *attributename3* specify the names of attributes 1, 2, and 3, respectively. *valuelist1*, *valuelist2*, and *valuelist3* are of the format:

*value1:value2:value3:...valuen*

To use fewer than three attributes, omit the appropriate attribute and value lines. Attributes must always be numbered sequentially, starting at 1. For example:

```
[iw_workflow_ui]
attribute1=Category
values1=Sales:Marketing:Engineering:Professional Services:Administration
attribute2=Month Due
values2=Jan:Feb:Mar:Apr:May:Jun:Jul:Aug:Sep:Oct:Nov:Dec
attribute3=Date Due
values3=1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24
:25:26:27:28:29:30:31
```

would create three attributes named Category, Month Due, and Date Due. Category has possible values of Sales, Marketing, Engineering, Professional Services, and Administration. Month Due has possible values of Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, and Dec. Date Due can be any number from 1 to 31.

**Note:** Due to page constraints, some lines may appear to wrap. Actual lines in `iw.cfg` should never wrap.

## Configuring Email Settings

The TeamSite Assign feature sends email to the recipient of a task. The following settings in the `[iwsend_mail]` section of `iw.cfg` enable you to configure how this email is sent:

- `maildomain=domain.topleveldomain`  
Specifies the domain (for example, `maildomain=example.com`)
- `mailserver=servername.domain.topleveldomain`  
Specifies the mail server to use.
- `use_mapping_file=false|true`  
Optional entry that specifies whether or not to use a mapping file to configure individual email addresses or aliases.
- `email_mapping_file=path_to_file`  
Optional entry that specifies the location of the mapping file to use (a sample file is located in `iw-home/local/config/wft/email_map.cfg`).

# Configuring Server Functionality

## Specifying the Encoding of the `iw.cfg` File

To facilitate the internationalization of TeamSite, you now have the ability to use text editors that save the `iw.cfg` file in various encodings. The encoding setting in the first section of the `iw.cfg` file declares the encoding of the `iw.cfg` file itself. The default setting is `ascii` and specified using the following syntax:

```
[iwcfg]
encoding=ascii
```

**Note:** This *must* be the first line in the `iw.cfg` file—no other entry can precede it.

You can edit the encoding setting using any of the following values:

| Preferred Encoding                   | Also Valid                                                          |
|--------------------------------------|---------------------------------------------------------------------|
| <code>ascii</code> (default setting) | <code>us-ascii</code>                                               |
| <code>ISO-8859-1</code>              | <code>iso_8859-1</code> and <code>latin1</code>                     |
| <code>windows1252</code>             | <code>cp1252</code>                                                 |
| <code>euc-jp</code>                  | <code>euc_jp</code>                                                 |
| <code>shift_jis</code>               | <code>shift-jis</code> , <code>sjis</code> , and <code>cp932</code> |
| <code>utf-8</code>                   | <code>utf_8</code> and <code>utf8</code>                            |

**Note:** You cannot have a space between the equal sign (=) and the encoding type.

## User and Role Authentication

### User Authentication

TeamSite can be configured to authenticate users by the following methods:

- LDAP—Users' credentials are passed to an LDAP server for verification. This requires an external LDAP server which can also be used for TeamSite role authentication.

- External file—Users' credentials are checked against a customer-specified file that is in the same format as /etc/passwd or a shadow password file. The file name is specified in the [authentication] section of `iw.cfg` using the `password_file` parameter.
- Local—Users' credentials are passed to the Solaris user authentication. If there is a shadow password file, that is used to verify the credentials. If no shadow password file is available, use the /etc/passwd file (this is the default method).
- Pluggable authentication modules (PAMs)—Users' credentials are passed to a PAM for verification (Solaris 2.6 or later)

Complete the following procedure to specify the type of authentication used by TeamSite:

1. Add the following line to the [authentication] section of `iw.cfg`:

`authenticate_by=mode`

where `mode` specifies one or more of `file`, `ldap`, `local`, or `pam`.

You can separate multiple entries by commas or spaces, and you can specify precedence. For example, if you specify `authenticate_by = file pam`, the `file` is checked first. If this check fails, `pam` is checked next.

When TeamSite roles are stored in LDAP, it is not possible to use multiple authentication methods: the `authenticate_by` parameter must be set to only `ldap`.

2. If `ldap` is specified in step 1, add the following lines to the [authentication] section of `iw.cfg`:

```
ldap_server=ldap-server
ldap_port=ldap-port
ldap_dnbase=search-base-location
ldap_key=key
```

where:

- `ldap_server` is the name or IP address of the LDAP server
- `ldap_port` is the port for the LDAP server (optional; the default value is 389)
- `search-base-location` is the specification of DN base location according to LDAP search syntax (for example, `ldap_dnbase=ou=people,o=example.com`)
- `key` is the name of the LDAP attribute that holds the user account names (optional; the default value is `uid`)

**Note:** If you are using LDAP, you must reset the TeamSite server with the `iwreset` command after you make changes to the LDAP database. This command causes the TeamSite server to replace existing user information in its local cache with new data from the LDAP server.

### Role Authentication Using LDAP

TeamSite role information is often stored separately from user authentication data. However, TeamSite enables you to store role information in an LDAP database along with user authentication data.

**Note:** Placing TeamSite roles in your LDAP server means that these roles can not be queried or manipulated programmatically using Interwoven's OpenAPI interface. TeamSite roles within your LDAP server can be queried and manipulated directly via your LDAP tool or programmatically via an LDAP server programming interface supplied by other vendors, not by Interwoven.

Every LDAP directory has a schema which describes the objects and attributes that are found in the directory. For example, you could have an object called `user` and an attribute `postaladdress`. To configure TeamSite to perform user and role authentication, you can either modify an existing attribute to represent TeamSite roles or create a new one.

For more information about modifying LDAP schemas, see “Modifying LDAP Schemas to Store TeamSite Roles” on page 150.

If you want to use your LDAP database to authenticate TeamSite roles (Author, Editor, Administrator, Master), add the following line to the `[authentication]` section:

`ldap_roles=role-attribute-name`

where `role-attribute-name` is the attribute name from the LDAP schema that stores TeamSite roles.

Additionally, you must modify your LDAP schema as described in “Modifying LDAP Schemas to Store TeamSite Roles” on page 150.

3. If `file` is specified as one of the possible modes (in step 1), add the following line to the `[authentication]` section:

`password_file=absolute-path-to-file`

where *absolute-path-to-file* is the absolute path to a file containing encrypted user-passwords of the same format found in `/etc/shadow`.

4. If `pam` is specified as one of the possible modes (in step 1), you can perform additional PAM-specific configuration activities as described in “PAM and Account Management” and “TeamSite for Solaris and PAM Configuration File Interaction” on page 151.

### ***Configuring TeamSite and LDAP to Work Without Using an Anonymous Bind***

In some installations, anonymous binds to LDAP are not permitted for security reasons. If you cannot use an anonymous bind to LDAP to read user names, you can establish a dedicated LDAP user account to use for user authentication. All searches for users' Distinguished Names will be done using this account instead of an anonymous bind. For this mode of operation, you must add two additional parameters to the `[authentication]` section of `iw.cfg`.

```
ldap_account=DistinguishedName
ldap_pwd=password
```

where:

*DistinguishedName* specifies the Distinguished Name of the LDAP user account to be used for LDAP searches. Note that this is not a simple account name, but a complete Distinguished Name (DN) for a user. For example:

```
ldap_account=cn=TeamSite,cn=Users,ou=myCompany,c=us.
```

*password* specifies the clear text password of the LDAP user account that matches `ldap_account`.

Note that the user name and password are in clear text, so it is important to limit who has read permission for `iw.cfg`. You can change the account name and password at any time. The changes will take effect the next time the server is restarted or reset.

## ***Modifying LDAP Schemas to Store TeamSite Roles***

If you do not have an existing attribute in your LDAP schema where you can store TeamSite roles, add a new attribute to your LDAP schema. If you do have an attribute where you can store TeamSite roles, start with step 3.

1. Add an auxiliary class to an existing object in the schema.
2. Add a new attribute to that object named *tsrolesattribute*.
3. Edit the [authentication] section of *iw.cfg* to include:  
`ldap_roles=tsrolesattribute`
4. Your LDAP administrator can now assign TeamSite roles (Master, Administrator, Editor, Author) to users configured in your LDAP server using the server's administration tools.

The following are the valid values (they are case sensitive):

- master
- admin
- editor
- author

5. Save and close the file.

### **Notes:**

- Placing TeamSite roles in your LDAP server means that these roles can not be queried or manipulated programmatically via the OpenAPI interface. TeamSite roles within your LDAP server can be queried and manipulated directly via your LDAP tool or programmatically via an LDAP server programming interface supplied by other vendors, not by Interwoven.
- You cannot store TeamSite role information in your LDAP database if you want to use operating system authentication. If you want to store role information in your LDAP database, also use LDAP for authentication.
- For information on modifying schemas and adding attributes for the Netscape Directory Server, refer to the *Directory Server Administrator's Guide* (<http://home.netscape.com/eng/server/directory>).

## PAM and Account Management

TeamSite for Solaris (TeamSite for AIX does not support PAM) communicates with pluggable authentication modules to perform account management functions on the authenticated user. This is typically used to control expired passwords, login time restrictions, etc. To configure TeamSite not to perform account management functions, add the following line to the [authentication] section of *iw.cfg*:

```
pam_do_acct_mgmt=no
```

## TeamSite for Solaris and PAM Configuration File Interaction

By default, on TeamSite for Solaris, PAM will control authentication by using entries tagged with the `teamsite` service name in */etc/pam.conf*. You can specify that PAM use */etc/pam.conf* entries tagged other than `teamsite` by changing the `pam_service` line in the [authentication] section of *iw.cfg*. For example, to specify that TeamSite instead use the lines in */etc/pam.conf* that also control the `telnet` program, edit the `pam_service` line in *iw.cfg* so that it reads as follows:

```
pam_service=telnet
```

The format of */etc/pam.conf* is described in detail in the `pam.conf(4)` man page. You should configure TeamSite with its own entries in */etc/pam.conf* using the service name `teamsite` (or whatever service name you specify for `pam_service` in *iw.cfg*). Only the auth and account modules in */etc/pam.conf* are used for TeamSite authentication. If no entries are present for TeamSite in */etc/pam.conf*, PAM will use whatever settings are specified for the other service. Note that this scenario is not recommended.

On Solaris 2.7 or later, the following lines in */etc/pam.conf* will produce behavior equivalent to the traditional TeamSite authentication method:

```
teamsite auth required /usr/lib/security/pam_unix.so.1
teamsite account required /usr/lib/security/pam_unix.so.1
```

On Solaris 2.6 systems, you must add the `use_first_pass` flag as follows:

```
teamsite auth required /usr/lib/security/pam_unix.so.1 use_first_pass
teamsite account required /usr/lib/security/pam_unix.so.1
```

To use a third-party PAM module, specify its path instead of `/usr/lib/security/pam_unix.so.1`. For more information about PAM, see:

[http://www.sun.com/software/solaris/pam/.](http://www.sun.com/software/solaris/pam/)

## Webserver UID

The webserver uid setting should be set to any uid that allows the webserver to see the web content as an outside viewer would see it, in order for users to be able to preview the Web site that a normal user would see. Because external browsers access the web server as `nobody`, this is used as the default. To change the webserver uid setting, edit the `webserver_uid` line in the `[iwserv]` section of `iw.cfg`. If `iw.cfg` does not contain this line, add it as shown below:

```
webserver_uid=nobody
```

## Web Daemon

To set Web daemon defaults, edit the `[iwwebd]` values in the `iw.cfg` file:

```
[iwwebd]
host=hostname.domain
http_port=80
https_port=443
default_protocol=http
```

The `default_protocol` setting is used by the following scripts when TeamSite generates URLs:

- `iwsend_servlet_mail.ipl` script—uses it to embed URLs into the email messages it sends
- `<iwov_webdesk_url>` presentation template tag—uses it when generating hyperlinks to the TeamSite server (see the *TeamSite Templating Developer's Guide* for more information)

## Servlet Engine

By default, the servlet port is set to 8080. To change this setting, edit the `servlet_port` value in the `[teamsite_servlet_ui]` section of the `iw.cfg` file.

```
[teamsite_servlet_ui]
servlet_port=8080
```

## Main Branch Settings

The following settings apply only to the main branch in TeamSite, not to any of its sub-branches. Because the main branch is not ordinarily used for development, these settings may not apply to your TeamSite configuration. However, if you have a special need to change the locking model, owner, or group of the main branch, you can use the following settings.

### Locking Model

TeamSite allows you to specify the locking model of each branch at the time that it is created. However, the main branch is created automatically when TeamSite is installed, or when a new backing store is created. You can specify which locking model to use for the main branch of a new backing store by editing the `main_lock_model` line in the `[iwserver]` section of `iw.cfg` (for a detailed explanation of locking models, refer to the *TeamSite User's Guide*). When TeamSite is first installed, it uses the default option of Submit locking for the main branch. The type of locking a branch uses cannot be changed after the branch has been created. However, if you edit the `main_lock_model` line and then create a new backing store, the new settings will take effect on the new backing store. For information about creating a new backing store, see page 242.

```
main_lock_model=locking_model
```

where `locking_model` is one of `submit_lock`, `optional_write_lock`, or `mandatory_write_lock` (or, more simply, `s`, `o`, or `m`). Submit locking is the default option for the main branch. Optional and mandatory write locking may significantly reduce system performance.

### Owner and Group

You can specify the owner and group of the main branch of a new backing store by editing the `main_owner` and `main_group` lines in the `[iwserver]` section of `iw.cfg`. When TeamSite is first installed, it uses the default option of `root` for main branch ownership. To change this setting on an existing main branch, you must use the `chown` and `chgrp` commands to change the ownership of the root directory of the main branch. However, if you edit the `main_owner` and `main_group` lines and then create a new backing store, the new settings will take effect on the new backing store. For information about creating a new backing store, see page 242.

```
main_owner=root
main_group=root
```

## Locked File Submission

You can configure TeamSite to allow only the owner or creator of the lock to submit a locked file to the staging area (as opposed to allowing any member of the workarea where the file is locked). To configure this option, add the following line to the [iwserver] section of `iw.cfg`:

```
only_lock_owner_creator_submits=yes
```

## Submit and Update Logs

You can configure the number of events that are contained in the Submit and Update logs for a workarea. To change this number, remove the comment (#) symbol from the `event_log_size` line in the [iwserver] section of `iw.cfg` and edit the line to specify the number of events you want to record. If this line does not appear in `iw.cfg`, add it as shown below. For example, with this setting, the Submit and Update logs will contain the 64 most recent Submit or Get Latest operations (as opposed to the 64 most recent files that were submitted or updated).

```
event_log_size=64
```

You can also configure whether or not you want all the files contained in new or deleted directories to be listed individually in the Submit and Update logs. To configure this option, remove the comment (#) marks from the `full_submitlog` and `full_updatelog` lines in the [iwserver] section of `iw.cfg` and edit the lines to specify yes to show all the files that were contained within the directory that was added or deleted or no to show only the directory names. If these lines do not appear in `iw.cfg`, add them as shown below.

```
full_submitlog=no
full_updatelog=no
```

## Branch and Workarea Security

Branch and workarea security determines whether or not a user can see the names of branches and workareas he does not have access to. If a user does not have read access to a branch or workarea, and branch and workarea security is turned off, he will be able to see the name of the branch or workarea, but it will not be linked, and [N/A] will appear next to it. However, you can configure TeamSite to not even show the names of branches and workareas in the TeamSite GUI if the user does not have read permissions. To set this option, remove the

comment (#) marks from the `branch_security` and `workarea_security` lines in the `[iwserver]` section of `iw.cfg` and edit the lines to specify `off` to show all branch and workarea names or `on` to show only the branch and workarea names for which the user has read access.

If these lines do not appear in `iw.cfg`, add them as shown below.

```
branch_security=on
workarea_security=on
```

## Default Permissions

You can configure the default permissions for branches, workareas, directories, and files created using TeamSite's GUI. Permissions on files created through the file system interface are determined by your file system interface configuration (for example, the Samba configuration).

To set the permission bits automatically, edit the `branch_default_perm`, `workarea_default_perm`, `directory_default_perm`, and `file_default_perm` lines in the `[iwserver]` section of `iw.cfg` to specify the octal values of the default permission bits for newly created branches, workareas, directories, and files. These settings will only apply to branches, workareas, directories, and files created after you have edited these lines. If these lines do not appear in `iw.cfg`, add them as shown below.

```
branch_default_perm=permissions
workarea_default_perm=permissions
file_default_perm=permissions
directory_default_perm=permissions
```

where `permissions` specifies the permissions in octal. For example:

```
branch_default_perm=775
```

## Group Remapping

This option provides a workaround for a limitation of NFS. When NFS checks the group that can access a file against the groups that a user belongs to, it only checks the first sixteen groups that the user belongs to. Therefore, if the group on the file is the seventeenth (or more) group that the user belongs to, NFS will incorrectly deny the user access to the file (this applies only to operations performed through the file system).

The `map_secondary_to_primary` option in `iw.cfg` works around this problem. Because TeamSite does not have NFS's sixteen-group limitation, it first determines whether a user should have group-level access to the file. Then, if the `map_secondary_to_primary` option is turned on, it maps the file's group to the user's primary group. Therefore, if you check the gid on a file via the file system, it could return a gid different from the true gid of the file. To find the file's true gid, use the **File > File Properties** menu item in the TeamSite GUI, or the `iattrib` command-line tool.

To turn group remapping on, add the following line to the `[iwserv]` section of `iw.cfg`:

```
map_secondary_to_primary_gid=yes
```

## File Locations

The `[locations]` section of `iw.cfg` may be used to change the locations of various TeamSite files and directories. To change the location of one of the following files or directories, remove the comment (#) marks from its line and edit the line to point to the new location (ensure that the `[locations]` line is not also commented out). After restarting, TeamSite looks for the specified file or directory in the new location.

If you change the location of `iwmount`, you will need to edit its webserver alias to point to the new location. In addition, any existing files in `/etc/defaultiw*` take precedence over these settings.

If you change the location of one of the logs, and no file of the specified name is present in the new location, a new file will be created.

If `iw.cfg` does not contain these lines, add the ones you want to configure as shown.

```
[locations]
iwbin=/usr/iw-home/bin
iwmount=/iwmnt
iwcgimount=/.iwmnt
iwroles=/usr/iw-home/local/config/roles
iwstore=/local/iw-store
ibsubmitconfig=/usr/iw-home/local/config/submit.cfg
iawuthorize=/usr/iw-home/local/config/autoprivacy.cfg
iwlogs=/usr/iw-home/local/logs
iwconfigs=/usr/iw-home/local/config
iweventlog=/var/adm/iwevents.log
iwtracelog=/var/adm/iwtrace.log
iwservlog=/var/adm/iwserv.log
iwdploylog=/usr/iw-home/local/logs/iwdploy.log
launchpad=iw-home/local/config/launchpad.cfg
```

where:

|                             |                                                                                                                              |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------|
| <code>iwbin</code>          | Specifies the location of TeamSite binaries (normally <code>iw-home/bin</code> ).                                            |
| <code>iwmount</code>        | Specifies the location of the TeamSite mount point.                                                                          |
| <code>iwcgimount</code>     | Specifies the location of the TeamSite mount point.                                                                          |
| <code>iwroles</code>        | Specifies the directory containing the TeamSite roles files.                                                                 |
| <code>iwstore</code>        | Specifies the location of the TeamSite backing store.                                                                        |
| <code>ibsubmitconfig</code> | Specifies the location of the Submit Filtering configuration file.                                                           |
| <code>iawuthorize</code>    | Specifies the location of the Autoprivacy configuration file.                                                                |
| <code>iwlogs</code>         | Specifies the directory containing TeamSite logs.                                                                            |
| <code>iwconfigs</code>      | Specifies the default configuration file directory.                                                                          |
| <code>iweventlog</code>     | Specifies the location of the TeamSite event log.                                                                            |
| <code>iwtracelog</code>     | Specifies the location of the TeamSite trace log.                                                                            |
| <code>iwservlog</code>      | Specifies the location of the TeamSite server log.                                                                           |
| <code>iwdploylog</code>     | Specifies the location of the deployment log.                                                                                |
| <code>launchpad</code>      | Specifies the location of the LaunchPad autoconfiguration file (default is <code>iw-home/local/config/launchpad.cfg</code> ) |

## Autoprivate

TeamSite's Autoprivate feature allows you to prevent certain file types and directories, such as temporary files and Macintosh resource forks, from being submitted to the staging area or copied during a **Copy To** operation. File types specified in the Autoprivate configuration files automatically get marked Private. For more information about Private files, see the *TeamSite User's Guide*.

**Note:** Changes to Autoprivate only apply to files or directories that are created or renamed after the changes are made. Changes do not apply to existing files.

To turn on Autoprivate, create a file named `autoprivate.cfg` in your `iw-home/local/config/` directory. The Autoprivate file consists of two sections:

- files (or directories) matched by pattern
- files (or directories) matched by name

Each section is set off by parentheses on their own lines, and the file begins with a “( ” (open parenthesis) on its own line and ends with a “ ) ” (close parenthesis) on its own line.

Individual entries in the first section are in the following format:

```
((filenamepattern)(#_characters_to_match_at_beginning)
(#_characters_to_match_at_end))
```

where both `#_characters_to_match_at_beginning` and `#_characters_to_match_at_end` are in hexadecimal.

For example, to have Autoprivate detect any file or directory that ends with `.frk`, add the following entry:

```
((x.frk)(0)(4))
```

meaning to match zero characters at the beginning of the name and the four characters (`.frk`) specified at the end of the name.

To set Autoprivate to detect any filename that ends in `.backup.fm`, add the following entry:

```
((x.backup.fm)(o)(a))
```

where `o` specifies not to match any characters at the beginning, and `a` (hexidecimal 10) specifies to match ten characters at the end of the filename.

Entries in the second section specify exact filename matches, set off by double parentheses. These filename matches apply across all directories in all workareas on the TeamSite server. For example, if `autoprivate.cfg` includes:

```
((test))
```

then all files and directories named `test` that are created after this line is added, in all directories in all workareas in TeamSite, will be marked private.

The `autoprivate.cfg` file recognizes the following six special characters: `( ) [ ] #` and a space (spacebar). If your file names contain any of these characters, you must encode these values when specifying them as a pattern. For example, to have Autoprivate detect a file name that includes spaces, encode the spaces with a `\20`, for example, to match “Network Trash Folder”:

```
((Network\20Trash\20Folder))
```

Encodings are represented as `\xx` where `xx` is the hex value of the corresponding ASCII character. The following table shows the mappings for the six special characters.

| Special Character | Autoprivate Encoding |
|-------------------|----------------------|
| #                 | \23                  |
| [                 | \5b                  |
| ]                 | \5d                  |
| (                 | \28                  |
| )                 | \29                  |
| space (spacebar)  | \20                  |

Encoding examples:

|                          |                                                     |
|--------------------------|-----------------------------------------------------|
| ((\23x\23)(1)(1))        | matches file names of the form: #*#                 |
| ((\23bbaax)(2)(0))       | matches: #b*                                        |
| ((\28ab\29)(2)(2)) #(ab) | matches the file name: (ab), the #(ab) is a comment |
| ((a\5b\5db)(2)(2))       | matches: a[ ]b                                      |

The following sample `autoprivate.cfg` file includes a few common entries:

```
(
(
((x.o)(0)(2))
((x.a)(0)(2))
((x~)(0)(1))
((.nfsXXX)(4)(0))
((x.bak)(0)(4))
((x.tmp)(0)(4))
)
(
((network\20trash\20folder))
((Network\20Trash\20Folder))
((.HSAncillary))
((.HSResource))
((.hsancillary))
((.hsresource))
((.tnatr:intf))
((.tnatr:reso-fork))
((resource.frk))
((trash))
)
)
```

For changes to `autoprivate.cfg` to take effect, restart the TeamSite server or use the `iwreset` command-line tool.

## New File Templates

If you are using templates with TeamSite's New File feature, you can configure which templates are to be used in various parts of the Web site. These settings are controlled through the templating configuration file, `iw-home/local/config/iwtemplates.cfg`.

This file governs which templates are accessible from which directories and branches. To configure which directories a template can be used in, add a line to this file. Only non-TeamSite Templating access is configured in this file (for example, HTML templates or Microsoft Word templates). To configure TeamSite Templating, consult the TeamSite Templating manual.

### Syntax

`iwtemplates.cfg` uses the following format:

```
templates
{
backing_store/main/branch
{
template_type
{
template_identifier
{
template=regular-expression
}
}
}
}
```

where:

- *backing\_store/main/branch* specifies the vpath to a branch and backing store in a MultiStore environment. If you are using a single backing store, *branch* specifies the vpath to a branch.
- *template\_type* specifies the type of template to be configured in that section.
- *template\_identifier* is the individual template identifier that will appear in the New File GUI.
- *template=regular-expression* specifies the full path of each template (rooted in a workarea) and uses case-insensitive regular expression matching to determine which directories this template may be accessed from.

You can have any number of *branch* sections, each of which can have any number of *template\_type* sections. The *template\_type* sections can have any number of *template\_identifier* sections, each of which contains one *template* line.

Each *template* line has the following format:

*template=regular-expression*

or

*template={regular-expression1, regular-expression2, regular-expression3...}*

The left-hand side of each line specifies the template, and the right-hand side contains the case-insensitive regular expression that determines where this template may be used.

Here are some common examples:

To permit the template `global.html` to be used across the entire Web site on a branch, you would add this *template* line to a branch section in `iwtemplates.cfg`:

```
/templates_dir/global.html=''
```

To permit the template `subdir.html` to be used in any directory path that contains a subdirectory named `subdir`, for example, `/htdocs/subdir/company` or `/htdocs/products/subdir`, you would add this line:

```
/templates_dir/subdir.html='subdir'
```

To permit the template `company.html` to be used in `/htdocs/company` and all of its subdirectories, you would add this line:

```
/templates_dir/company.html='^/htdocs/company/'
```

To permit the template `products.html` to be used in the `/htdocs/products/` directory but not its subdirectories, you would add this line:

```
/templates_dir/products.html='^/htdocs/products/$'
```

You can also specify multiple patterns to be matched. If you specify multiple patterns, enclose the right-hand side of the line in curly brackets and separate the individual patterns by commas. For example:

```
/templates_dir/products.html={'^/htdocs/products/' , 'subdir'}
```

would permit the `products.html` template to be used in `/htdocs/company` and all of its subdirectories, and in any directory path that contains a subdirectory named `subdir`.

## Launching Files Through iwproxy

This option enables in-context QA and consistent views of TeamSite workareas. By default, this option is turned on. However, if your Web site must support SSL, you will need to turn this option off and install TeamSite's redirector module. To install and configure the redirector module, see "Installing the Redirector Module for NES and iPlanet" on page 57(for Netscape web servers) or page 59 (for Apache web servers).

To turn this option off:

1. If a comment symbol (#) is present at the beginning of the `use_iwproxy` line of `iw.cfg`, remove it. If `iw.cfg` does not contain this line, add it as shown below.
2. Edit the line to read:  
`use_iwproxy=no`

## Configuring the TeamSite Server Locale

The `iw.cfg` file now contains a `server_locale` entry in the `[iwserver]` section. The entry specifies the locale in which current execution of the TeamSite server (`iwserver`) runs. For example:

```
[iwserver]
:
server_locale=English_UnitedStates.US-ASCII@Binary;
```

This setting is automatically written to the `iw.cfg` file when `iwserver` is started. The native locale is determined by reading the `LANG` environment variable Once the `server_locale`

setting exists in the `iw.cfg` file, it is used to determine the TeamSite server's native locale at every invocation of `iwserver`. If this setting is not present, `iwserver` determines its locale from the `LANG` environment variable.

**Note:** While this setting can be user-modified, it is designed to serve as reference as to the locale in which `iwserver` is currently running. If you have a situation where you want to force `iwserver` to run in a particular locale (independent of the `LANG` environment variable) you can manually set the `server_locale` field.

The locale in which the server operates (as defined by the `server_locale` entry), effectively determines the locale of the TeamSite IFS. For example, if `iwserver` runs under the `Japanese_Japan.Shift_JIS@Binary` locale, all file and directory names are encoded in `Shift_JIS` encoding.

The `server_locale` setting in the `iw.cfg` file can contain any of the locales listed in the following table (note that these settings are Interwoven naming conventions—the operating system locales to which they map are also contained in the table):

| <b>iw.cfg server_locale Setting</b>               | <b>Solaris 2.7 Locale</b>   | <b>AIX 5.1 Locale</b>       |
|---------------------------------------------------|-----------------------------|-----------------------------|
| <code>Japanese_Japan.Shift_JIS@Binary</code>      | <code>ja_JP.PCK</code>      | <code>Ja_JP.IBM-932</code>  |
| <code>Japanese_Japan.JapanEUC@Binary</code>       | <code>ja</code>             | <code>ja_JP</code>          |
| <code>German_Germany.Latin1@Default</code>        | <code>de</code>             | <code>de_DE</code>          |
| <code>French_France.Latin1@Default</code>         | <code>fr</code>             | <code>fr_FR</code>          |
| <code>English_UnitedStates.UTF-8@Binary</code>    | <code>en_US.UTF-8</code>    |                             |
| <code>English_UnitedStates.US-ASCII@Binary</code> | <code>C ("C" locale)</code> | <code>C ("C" locale)</code> |

## Configuring Server Performance

### Cache Size

To set TeamSite's cache size, edit the `cachesize` line in the `[iwserver]` section of `iw.cfg`. If a comment symbol (#) is present at the beginning of this line, remove it. If this line does not appear in your `iw.cfg` file, add it as shown below. The initial cache size setting should be approximately three times the number of files and directories on the largest branch.

For example, if the largest branch contains 15,000 files and directories, you should set cache size to 45000 as follows:

```
cachesize=45000
```

Minimum cache size is 1000; maximum is 400000 (four hundred thousand) entries. Each cache line takes a maximum of 1KB of physical memory. Recommended physical memory is cache size times 1KB plus an additional 25% as a safety margin. In the example shown below, physical memory would be  $(45,000 * 1KB) + 11MB = 56MB$ . If you encounter a great deal of memory swapping, you should either reduce the cache size or install more memory.

You must restart the TeamSite server for these changes to take effect.

## RPC Threadcount

The RPC threadcount setting determines how many simultaneous requests TeamSite can handle from users via the GUI or command-line tools. These requests are very short-lived, so that threads are quickly freed for other users. If all threads are currently being used, TeamSite starts to serialize requests. This setting should not be altered.

```
rpc_threadcount=64
```

## File System Threadcount

The file system threadcount should be set to approximately the number of CPUs on the TeamSite server. To change the file system threadcount, edit the `fs_threadcount` line in the `[iwserver]` section of `iw.cfg`. If a comment symbol (#) is present at the beginning of this line, remove it. If this line does not appear in your `iw.cfg` file, add it as shown below.

```
fs_threadcount=2
```

You must restart the TeamSite server for this change to take effect.

## Filesystem Active Area Cache

The file system active area cache should be set to approximately the number of users who are expected to be using TeamSite concurrently. Note that this is the number of users who are using TeamSite at one time, not the total number of TeamSite users. If this value is too large, it will significantly impact memory usage.

To set the file system active area cache, edit the `fs_active_area_cache` line in the `[iwserver]` section of `iw.cfg`. If a comment symbol (#) is present at the beginning of this line, remove it. If this line does not appear in your `iw.cfg` file, add it as follows:

```
fs_active_area_cache=8
```

You must restart the TeamSite server for this change to take effect.

## Throughput Monitors

Throughput monitors can be used in conjunction with the `iwstat` command-line tool to monitor system status and performance. To turn on throughput monitors, remove the comment marks (#) from the beginning of the lines for the throughput monitors you want to use in the `[iwserver]` section of `iw.cfg`. By default, there are throughput monitors that return system statistics over the previous minute, fifteen minutes, hour, 8 hours, 24 hours, and for the entire time that the system has been running. There are also two throughput monitors that you can configure with any time interval.

```
thruputmonitoring=on
thruputmonitor1=1 # 1 minute
thruputmonitor2=15 # 15 minutes
thruputmonitor3=60 # 1 hour
thruputmonitor4=480 # 8 hours
thruputmonitor5=1440 # 24 hours
thruputmonitor6=-1 # forever
thruputmonitor7
thruputmonitor8
```

## Detecting Low Disk Space and Inode Count

TeamSite is configured to freeze the backing store when it detects that free disk space or inode count is low. The backing store remains frozen until sufficient disk space or inode count is restored, at which point the server returns to its normal running state. This feature helps prevent possible corruption of the backing store. While the backing store is frozen, users cannot write to the TeamSite backing store. Users can still perform read-only operations. The CLT `iwfreeze` can be used to manually freeze the backing store.

The lines shown below in the `[iwserver]` section of `iw.cfg` control the behavior of disk/inode low detection.

The `disklow_mbytes` line gives the server a freeze threshold in MB (the default is 50). The `disklowpercent` line sets the percent of free disk space that is considered “low” (the default is 10). The TeamSite server does not allow `disklowpercent` to go below 2%. If the server detects a low-disk state based on the threshold set in `iw.cfg`, it does not allow you to manually unfreeze the backing store via the `iwfreeze` command. The `disklow_knodes` line gives the server a freeze threshold in thousands of inodes (the default is 50). To change these settings, edit these lines as shown below.

```
disklow_mbytes=20
disklowpercent=15
disklow_knodes=25
```

## Submit Filtering

The TeamSite server allows you to automatically change file attributes, such as uid, gid, and permissions, at the time that you submit a file. This option allows you to automate the task of specifying the permissions that each file will have in the deployed Web site. The submit filter performs the specified operation on files immediately before they are submitted, so that changes are made to the files in the workarea, which are then submitted.

On startup, the TeamSite server reads a configuration file named `submit.cfg` in the `iw-home/local/config/` directory (unless the location of this file is otherwise specified in the `[locations]` section of `iw.cfg`). The `submit.cfg` file contains rules to match file and workarea patterns to specific actions to perform when files and directories are submitted.

It has the following format:

```
case-sensitive = [yes|no]
rules
{
 workarea1_pattern
 {
 file_pattern1 { action1 action2 ... }
 file_pattern2 { action3 action4 ...
 ...
 }
 workarea2_pattern
 {
 file_pattern3 { action5 action6 ... }
 file_pattern4 { action7 action8 ... }
 ...
 }
 ...
}
```

The case-sensitive statement specifies whether or not the rules matching should ignore the case of the path names. If case-sensitive is not specified, the value is assumed to be yes, so that rules matching does not ignore the case of the path names.

*workarea pattern* is used to match a workarea to the set of file rules to apply when a submit is done from the workarea. Each pattern can only be specified once, and the first match is used. The syntax of the pattern is `regex(5)` (extended syntax). For AIX, use the POSIX 1003.2 extended regular expression syntax. For more information on regular expressions, consult a reference manual such as *Mastering Regular Expressions*, by Jeffrey Friedl, or read the man page (Solaris only):

```
% man -s 5 regex
```

The match is done against the path name of the workarea, starting with /default/main.

*file pattern* is used to match a file or directory to the set of actions to perform on it when it is submitted. Each file or directory pattern can only be specified once, and the first match is used. The syntax of the pattern is `regex(5)` (extended syntax) for Solaris, or the POSIX 1003.2 extended regular expression syntax for AIX.

The match is done against the path name of the file or directory relative to the workarea.

*action* is one of

```
uid=name
gid=name
perm=octal number
amask=octal number
omask=octal number
expand_rcs_macros = [yes|no] (see page 172)
```

and specifies the operation to perform on the file or directory being submitted. *uid*=, *gid*=, and *perm*= specify new values for these attributes. *amask*= specifies a bit mask to “and” to the existing mode of the file or directory. *omask*= specifies a bit mask to “or” with the existing mode of the file or directory.

When you submit files or directories:

1. The server determines what files and directories have actually changed and need to be submitted. It also verifies that none of them are in conflict with the staging area or locked in other workareas.
2. The path name of the workarea from which the submit is being done is matched against the workarea patterns from the configuration file.
3. If the workarea matches one of the workarea patterns, then, for each file and directory that needs to be submitted (as determined in step 1), the file's path name is matched against the file patterns in the matching workarea's section.
4. If a match is found, then the server performs the specified set of actions to the file or directory in the workarea.
5. The server submits the transformed files and directories to the staging area.

### ***Example***

This is a sample `submit.cfg` file:

```
CASE-SENSITIVE = YES
RULES
{
SECTION 1
 ^/default/main/WORKAREA/.*$ # any workarea in the main branch
 {
 ^/index\.html$ { gid=test perm=0664 uid=andre }
 # attributes fixed for /index.html
 .*\.sh$ { omask=0111 } # execute bits on all .sh files
 /$ { uid=andre } # andre owns all the directories
 .* { amask=0775 } # don't allow world write access
 }
SECTION 2
 ^/default/main/TeamSite/WORKAREA/chris$ # just for chris
 {
 ^/html/chris/.*$ { uid=chris gid=iw } # under /html/chris/
 .* { amask=0775 } # don't allow world write access
 }
SECTION 3
 .* # any other workarea on any other branch
 {
 .* { amask=0775 gid=test } # no world write access
 }
}
```

## Notes

Only the first match is applied. That is, the first match wins if multiple rules match the file or directory. `submit.cfg` should be ordered so that the most specific workarea patterns are closer to the top and the most specific file patterns are earlier in each section. You may need to duplicate some actions for them to apply to multiple rules.

For purposes of matching, the path name of a directory must end in a slash (“/”).

Single or double quotes around patterns are optional, but they must be used around workarea and file patterns that contain white space or other special characters, like #, {, }, =, or ,. Backslashes (\) are special characters when used within patterns surrounded by quotes; a backslash followed by any character is replaced by the single character. For example, to embed a single quote, double quote, or backslash in a pattern, precede the character with a backslash (\', \", or \\). Backslashes are not special characters in patterns that are not quoted.

Do not specify duplicate workarea patterns multiple times, duplicate file patterns multiple times within a workarea section, or the same action multiple times within a file rule.

Because of client-side attribute caching, the modes, uid, and gid may not appear differently in the workarea immediately after a submit. The correct attributes will appear after a sufficient time-out interval (usually about 30 seconds).

You cannot change the permissions for a symbolic link. The `perm`, `amask`, and `omask` actions will not be performed on a submitted symbolic link. The `uid` and `gid` actions will be performed on a submitted link, not on the actual file.

The permission values should be specified as an octal number (starting with a 0), between 0 and 0777.

Changes to `submit.cfg` do not take effect until the server is restarted or until you use `iwreset`.

### ***Debugging***

The CLT `iwtestcfg` (see *TeamSite Command-Line Tools*) can be used to find out which workarea and file pattern will be applied to a file at the time of submission:

```
% iwtestcfg /default/main/WORKAREA/andre/cgi/test.sh
```

Would return:

```
Matched area pattern "^/default/main/WORKAREA/.*"$
Matched file pattern ".*\.sh$"
Actions to do are:
omask=0111
```

Matched area pattern and Matched file pattern are the case-insensitive regular expressions found in `submit.cfg` that match the workarea and file. Actions to do are the actions (specified in `submit.cfg`) that will be applied to the file.

You can also get debugging information on the submit handling configuration printed to `iwtrace.log`, by adding the following line to the [server] section of `iw.cfg`:

```
debug_event_handler=yes
```

This will cause the server to print a configuration map of `submit.cfg` and to print which actions are performed as files are submitted.

## RCS Macro Expansion

TeamSite provides RCS-style macro substitution at submit time. These macros give you a way to embed version information into files, such as the file name, revision string, last modifier, when it was modified, and submit comments.

To use the macro facility, you must first add new rules to the `submit.cfg` configuration file, then manually insert the macros into files in your workarea. When the files are submitted, the TeamSite server replaces the macros with the appropriate information, leaving you with a rewritten file in your workarea and the staging area.

### *Macros*

The macros are a subset of those used in RCS (called “keywords” in RCS documentation). The following description is taken from the UNIX `co(1)` man page, modified for TeamSite semantics.

Strings of the form `$keyword$` and `$keyword:...$` embedded in the text are replaced with strings of the form `$keyword:value$` where `keyword` and `value` are pairs listed below. Keywords can be embedded in literal strings or comments to identify a revision.

Initially, the user enters strings of the form `$keyword$`. On submit, the server replaces these strings with strings of the form `$keyword:value$`. On subsequent submits, the value fields will be replaced automatically with updated values.

Keywords and their corresponding values:

`$Author$`

The login name of the user who last modified the file. Note that this is a standard RCS keyword and does not refer to the TeamSite Author role.

`$Date$`

The date and time the file was last modified.

**\$Header\$**

A standard header containing the path name of the file, the revision string, the date and time, the author. The path is relative to the area root.

**\$Id\$**

Same as **\$Header\$**, except that the filename is without a path.

**\$Log\$**

The comment supplied during submit, preceded by a header containing the filename, the revision string, the date and time when the file was last modified, and the author. The comment is either the submit comment supplied for the individual file or, if this is empty, the comment supplied for the all the files associated with the submit. Existing log messages are not replaced. Instead, the new log message is inserted after **\$Log:...\$**. This is useful for accumulating a complete change log in a file.

Each inserted line is prefixed by the string that prefixes the **\$Log\$** line. For example, if the **\$Log\$** line is `// $Log:tan.cc $`, RCS prefixes each line of the log with " // ". This is useful for languages with comments that go to the end of the line. The convention for other languages is to use a \* prefix inside a multiline comment. For example, the initial log comment of a C program is conventionally of the following form:

```
/*
 * Log
```

**\$Revision\$**

The revision string.

**\$Source\$**

The pathname of the file, relative to the root directory of the area.

These RCS keywords are ignored by TeamSite:

```
$Locker$
$Name$
$RCSfile$
$State$
```

The following characters in keyword values are represented by escape sequences:

| Character   | Escape Sequence |
|-------------|-----------------|
| end-of-line | \n              |
| \$          | \044            |
| \           | \\              |

### ***Enabling Macro Expansion***

To enable RCS macro expansion for a particular set of files, you must include the following action in the `submit.cfg` rules that apply to those files:

```
expand_rcs_macros=yes
```

Here is a sample `submit.cfg` file:

```
CASE-SENSITIVE = YES
RULES
{
 "." # any workarea
 {
 ".*\.[ch]$" { gid=iw, expand_rcs_macros=yes }
 # files ending in .c or .h
 ".*" { gid=iw } # all other files/directories
 }
}
```

# Configuring the TeamSite Web Daemon and Proxy Server

## About the TeamSite Web Daemon

TeamSite uses a Web daemon, `iwwebd`, to provide SSL support for the TeamSite browser GUI. Remote contributors can use TeamSite securely without having to establish a Virtual Private Network (VPN). This Web daemon also serves up the non-servlet-based parts of the TeamSite GUI.

Note: The log directory must be on a lockable file system, typically “local.”

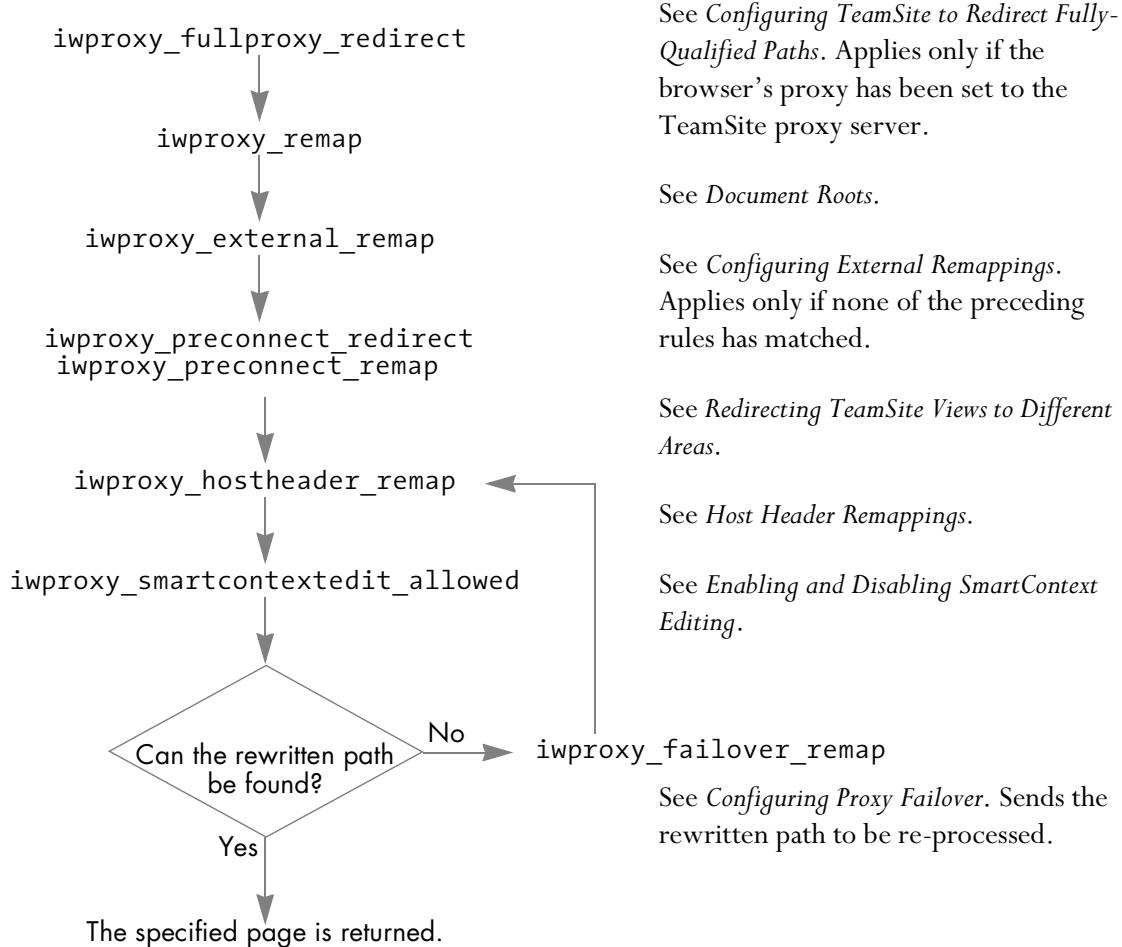
For an illustration of how requests are processed, see “TeamSite Architecture” on page 21.

## About the Proxy Server

TeamSite uses a proxy server to perform several important functions:

- Resolve relative and absolute URL names in TeamSite areas in order to present users with a virtualized view of the Web site contained within an area (see page 178).
- Redirect fully-qualified URLs into TeamSite areas (see page 182).
- Redirect browsing in one branch or workarea into another area (see page 186).
- Redirect individual workareas to use different Web servers (see page 189).
- Remap links to external Web servers (see page 190).
- Modify “Host:” headers (see page 191).
- Remap SSI requests (page 192).

Each time a request is made through the TeamSite proxy server, the following sections of `iw.cfg` are processed in the following order. More than one rule may be applied to a request. As a URL gets rewritten by a rule, the rewritten URL is passed to the next section. The first rule that matches in any section prevails; no other rule in that section will be applied.



## Applying Changes to Proxy Configuration

If you change the `iwproxy` mappings in `iw.cfg`, you will need to reset the server with the `iwreset -a` command-line tool to reflect the changes.

Note that `iwreset -a` will not apply changes to the `[iwproxy_remap]` or `[iwproxy_plugin_remap]` sections of `iw.cfg`, if you are using Web server plugins. If you make changes to these sections, and you are using webserver plug-ins, you will need to restart the Web server to apply the changes.

## Configuring TeamSite Web Daemon and Proxy Server Operation

The `[iwproxy]` section of `iw.cfg` is used to configure the operation of TeamSite's proxy server. For example:

```
[iwproxy]
iwproxy_port=1080
iwproxy_host=proxy_hostname
customer_webserver_port=81
customer_webserver_host=hostname
```

where:

`iwproxy_port` is the port TeamSite's proxy server will operate on. It should be set to an open port value (1080 is selected by default).

`iwproxy_host` specifies the host where the TeamSite proxy daemon runs. Usually this will be the TeamSite server.

`customer_webserver_port` is the port through which TeamSite's proxy server communicates with the Web server. It must be set to the value of the port used by the Web server. Port 81 is selected by default.

`customer_webserver_host` is the host name of the content Web server. The value must be set to the host name of the Web server that serves the content of your Web sites.

The settings in the `[iwproxy]` section are set during installation, and can be edited when necessary.

## Resolving Relative and Absolute Paths

### About Relative and Absolute Paths

Relative paths specify file locations relative to the referencing file's directory location. Absolute paths specify file locations relative to the Web site's document root directory. For example, the file whose directory path (rooted in a TeamSite area) is:

```
/main/index.html
```

might contain a link to the file

```
/images/banner.gif
```

This link can be specified as either a relative or an absolute path.

If the link were specified as a relative path, it would look like:

```
../images/banner.gif
```

If the link were specified as an absolute path, it would begin with a / and look like:

```
/images/banner.gif
```

**Note:** The proxy server does not allow you to remap the document root directory for backing store branches other than the default store.

### Resolving Relative and Absolute Paths

Links in HTML documents are often specified with relative or absolute path names. For example, in a link to an image, the file name might appear as:

```
/images/pic.gif
```

On a typical Web server, this link reference would be mapped by the Web server to its document root, for example:

```
/images/pic.gif ==> /usr/httpd/images/pic.gif
```

All users attempting to access the file using the absolute path name will be mapped to the same file location on the Web server.

However, TeamSite supports a system of private workareas, giving each user access to the Web site's files from within their own personal, virtual version of the Web site. TeamSite uses a proxy server to manage mapping of files to workareas with relative and absolute path references. Going back to our example, the TeamSite proxy server allows each user attempting to access `/images/pic.gif` from within TeamSite to be mapped to the copy of `pic.gif` in his own workarea. The redirected mapping would look like:

```
/images/pic.gif ==>
/iw-mount/default/main/branchpath/WORKAREA/workareaname/images/pic.gif
```

### **Document Roots**

TeamSite maps the initial Web server directory structure (*document root*) of workareas to the top level of the workarea directory by default. You may, however, want to move the document root, or group types of files together for improved clarity, convenience, or efficiency. On a branch-by-branch basis, the TeamSite proxy server allows you to remap the document root anywhere within the workarea directory. It also allows you to define mappings directly to sub-directories within workareas.

Path mappings are defined by including sections within the TeamSite main configuration file (`iw.cfg`).

To configure document roots for individual branches:

1. For each branch that you want to configure, add a line to the [iwproxy\_remap] section of `iw.cfg`, of the form:

`configsectionname=vpath`

where `vpath` is the vpath to the branch you are configuring, and `configsectionname` is the name of the section of the configuration file that will define the branch remappings.

2. For each line that you added to [iwproxy\_remap], create a section in `iw.cfg` named `[configsectionname]`. Add a line to this section that defines the document root:

`_docroot=dirpath`

where `dirpath` is a directory path rooted in a workarea.

You can also add lines that bypass the document root, of the format:

`path=path`

For example, you might add the following lines to [iwproxy\_remap]:

```
[iwproxy_remap]
branchrewrite_1=/main
branchrewrite_2=/main/training
```

The first line of the above example tells TeamSite to use the section [branchrewrite\_1] to set the document root configuration for the /main branch. The second line tells TeamSite to use the [branchrewrite\_2] section to set the document root configuration for the /main/training branch.

You would then create two new configuration file sections corresponding to the lines in [iwproxy\_remap]:

```
[branchrewrite_1]
_docroot=/htdocs
/pictures/=pictures/
/html/=html/
[branchrewrite_2]
_docroot=/htdocs
/images/=images/
```

The first line of both the new sections contains:

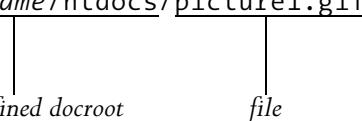
```
_docroot=/htdocs
```

This defines a special directive that sets the mapping of the document root. Any requests from workareas on the `main` branch or the `main/training` branch to the root level directory (/) will now start at:

```
.../workareaname/htdocs/
```

Thus, the request for file `/picture1.gif` will now be mapped to:

```
.../workareaname/htdocs/picture1.gif
```



The two docroot configuration sections also contain lines similar to the following:

```
/pictures/=pictures/
```

This line maps file requests directly to the listed directory `/pictures/`, bypassing the document root defined in the first line. Thus, a request for the file `/pictures/people.gif` gets mapped to:

```
.../workareaname/pictures/people.gif
```

not:

```
.../workareaname/htdocs/pictures/people.gif
```

TeamSite's proxy server operates using literal string matches and substitutions in path names. To avoid inadvertently rewriting names, always use trailing slashes in your remap definitions (but not your `_docroot` directories.)

**Note:** Do not use trailing slashes in your remap definitions for `_docroot` directories.

## Resolving Fully-Qualified URLs

TeamSite's proxy server can also be configured to resolve fully-qualified paths. For example, a link to the main page of a Web site might appear as

`http://www.name.com`

If such a link appears in an HTML file in a TeamSite workarea, and you follow that link while performing in-context QA, you will be taken out of the workarea and to the actual referenced Web site.

Therefore, if you use fully-qualified URLs to reference pages within your own Web site, clicking on these links will take you out of the in-context view of the current workarea, staging area, or edition and into your own currently deployed Web site. To solve this problem, TeamSite allows you to configure your proxy server. The proxy server will redirect fully-qualified links within your Web site, then pass them to the regular proxy server to ensure the integrity of the in-context view in a workarea, staging area, or edition.

**Note:** Only configure this setting if your Web site uses fully-qualified URLs that you need to view in-context! This setting requires you to manually configure your browser, so that you will not be able to view the actual Web site without reconfiguring your browser. Also, this slows the TeamSite server by sending every request through the Web daemon and iwproxy.

### Configuring TeamSite to Redirect Fully-Qualified URLs

To configure TeamSite to redirect fully-qualified URLs, you must:

- Configure the TeamSite proxy server.
- Set your (client) browser's proxy to the TeamSite Web daemon.

### *Configuring the TeamSite Proxy Server to Redirect Fully-Qualified URLs*

To configure the TeamSite server to redirect fully-qualified URLs, edit the [iwproxy\_fullproxy\_redirect] section of `iw.cfg`. This section contains any number of `_regex` lines. Each line is of the format:

`_regex=source_regex=dest_ex`

where *source\_regex* is a case-insensitive regular expression specifying a fully-qualified URL that might appear in a page, and *dest\_ex* is an expression specifying the path that the link will be redirected to. This expression should always be the path to the file specified in *source\_regex*, but rooted in a TeamSite area.

For example:

```
[iwproxy_fullproxy_redirect]
_regex=http://www(\.example\.com)?/(.*)=/\$2
```

redirects links that specify

`http://www.example.com`

in the URL and sends them to the corresponding location in the current TeamSite area.

**Note:** If your `iw.cfg` file's `[iwwebd]` section defines the host as `host=hostname.domain`, and your browser's proxy server is set to `hostname.domain:port`, when you start TeamSite, you must enter `http://hostname.domain/iw/` in your browser rather than `http://hostname/iw/`.

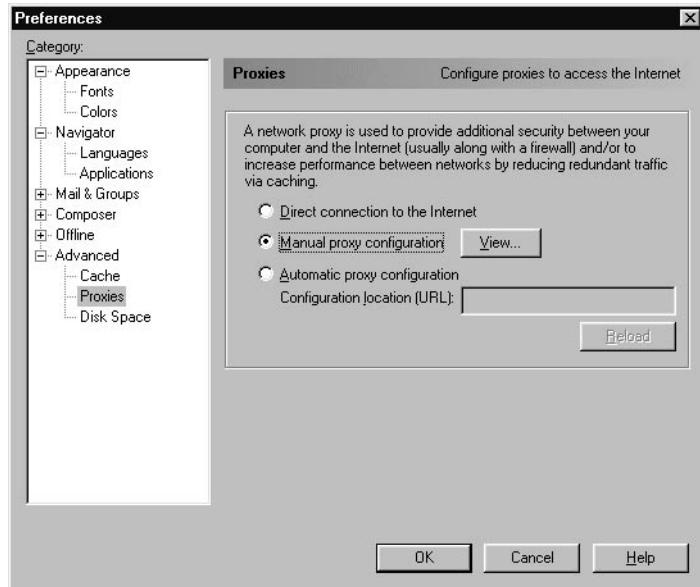
### *Configuring the Client for Fully-Qualified URL Redirection*

If you are using `[iwproxy_fullproxy_redirect]`, you must set up your (client) browsers to go through the TeamSite Web daemon. All requests will then go through `iwwebd`. If you need to browse one of the live Web sites that the TeamSite Web daemon reroutes requests for, you will need to set your browser to not use the TeamSite Web daemon.

#### **Netscape**

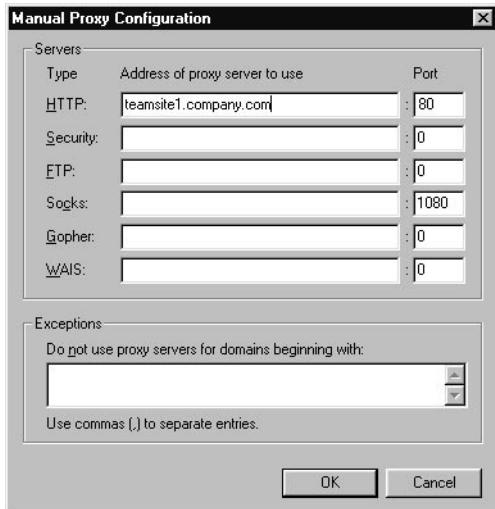
To configure your browser to use the TeamSite Web daemon:

1. In Netscape Navigator, select **Preferences** from the **Edit** menu.
2. The Preferences window will appear. Double-click on the **Advanced** category in the left-hand pane.
3. Select the **Proxies** subcategory in the left-hand pane.



*The Netscape Preferences window*

4. In the right-hand pane, select **Manual proxy configuration**. Select **View**.
5. The Manual Proxy Configuration window will appear. Type the name of your TeamSite server (for example `teamsite1.example.com`) in the **HTTP** section. Type the `iwwebd_port` specified in the `[iwwebd]` section of `iw.cfg` (for example, 80) in the **Port** section.



The Manual Proxy Configuration window

6. Click **OK**.
7. In the Preferences window, click **OK**.

To configure Netscape to not use the TeamSite proxy server:

1. In Netscape Navigator, select **Preferences** from the **Edit** menu.
2. The Preferences window will appear. Double-click on the **Advanced** category in the left-hand pane.
3. Select the **Proxies** subcategory in the left-hand pane.
4. In the right-hand pane, select **Direct connection to the Internet**.
5. Click **OK**.
6. In the Preferences window, click **OK**.

## Internet Explorer

To configure your browser to use the TeamSite proxy server:

1. In Internet Explorer, select **Internet Options** from the **View** menu. The Internet Options window will appear.
2. Select the **Connection** tab.
3. Select the **Access the Internet using a proxy server** checkbox.
4. Type the name of your TeamSite server (for example, `teamsite1.example.com`) in the **Address** section. Type the `http-port` specified in the [`iwwebd`] section of `iw.cfg` (for example, 80) in the **Port** section.
5. Click **OK**.

To configure Internet Explorer to not use the TeamSite proxy server:

1. In Internet Explorer, select **Internet Options** from the **View** menu. The Internet Options window will appear.
2. Select the **Connection** tab.
3. Deselect the **Access the Internet using a proxy server** checkbox.
4. Click **OK**.

## Redirecting TeamSite Views to Different Areas

TeamSite's proxy server allows web teams to work on branches of development that are populated only with the portion of the Web site that they are developing, but still maintain a fully in-context view of the entire Web site by referencing the staging area or a known edition on another branch of development.

This feature is very flexible in that it can be configured on a per-branch or per-workarea basis, and the redirected view can be configured to take the user to any TeamSite area on any branch. Redirection can occur in one of two ways:

1. Through [`iwproxy_preconnect_remap`], which retains your original area as the current working area and directs files there from another area. In this scenario, docroot is based on the original area's parent branch.
2. Through [`iwproxy_preconnect_redirect`], which causes the area you redirect into to become the current working area (and that area's parent branch becomes the basis of docroot).

### **Using [`iwproxy_preconnect_remap`]**

To configure TeamSite to redirect workarea views as described in Item 1 above, edit the [`iwproxy_preconnect_remap`] section of `iw.cfg`:

```
[iwproxy_preconnect_remap]
_regex=source_regex=dest_ex
```

where *source\_regex* is a case-insensitive regular expression describing the area to be mapped from, and *dest\_ex* is an expression describing the area to be mapped to. These areas are most commonly workareas or staging areas, but you can map to and from any workarea, staging area, or edition. You can add any number of *\_regex* lines to this section.

For example:

```
_regex=(.*)/branch1/WORKAREA/[^/]+/products/(.*)=$1/branch2/
STAGING/products/$2
```

tells the proxy server to remap the `products` directory of any workarea on any branch named `branch1` to the `products` directory of the staging area on its sister branch, `branch2`.

In the source regular expression, `( .*)` is used to specify an arbitrary path, and `$1` in the destination expression means that it must follow the same path (and therefore `branch1` can be anywhere in the branch structure, but `branch2` is a sister branch of `branch1`). Also in the source regular expression, `[ ^/ ]+` is used to specify a single directory level, of any name (which in this case would be the workarea name, and therefore all workareas on `branch1` are specified).

Finally, the source regular expression uses `(.*)` to specify another arbitrary path, and `$2` in the destination expression tells it to follow the same path.

You can also specify the exact location of the areas you want to remap:

```
_regex=^/
iw-mount/default/main/branch1/WORKAREA/[^/]+/products/(.*)=/
iw-mount/default/main/branch2/STAGING/products/$1
```

Or, you can specify an individual workarea to remap:

```
_regex=^/
iw-mount/default/main/dev/branch1/WORKAREA/andre/coolstuff/(.*)=/
iw-mount/default/main/branch2/STAGING/coolstuff/$1
```

The TeamSite proxy server applies the first match it finds, so you can exclude a particular area from a more general rule by creating a separate rule for that area and placing it before the more general rule. For example:

```
_regex=(.*)/branch1/WORKAREA/chris/products/(.*)=$1/branch1/
STAGING/products/$2
_regex=(.*)/branch1/WORKAREA/[^/]+/products/(.*)=$1/branch2/
STAGING/products/$2
```

remaps the `products` directory in all workareas on `branch1` except for Chris's to the staging area of `branch2`.

See “Configuring Proxy Failover” on page 192 for a details about configuration rule precedence.

## Using [iwproxy\_preconnect\_redirect]

To configure TeamSite to redirect workarea views as described in Item 2 above, edit the [iwproxy\_preconnect\_redirect] section of `iw.cfg`:

```
[iwproxy_preconnect_redirect]
_regex=source_regex=dest_ex
```

where `source_regex` and `dest_ex` are as described in “Using [iwproxy\_preconnect\_remap]” on page 187. If you set [iwproxy\_preconnect\_redirect] and then click on a link defined by an absolute path name, the docroot of that link is based on the branch you redirected into (as opposed to the branch of the area you redirected from, which would be the behavior if you had set [iwproxy\_preconnect\_remap]). See “Configuring Proxy Failover” on page 192 for a details about configuration rule precedence.

## Configuring TeamSite to Use Different Web Servers

You can configure TeamSite to use different Web servers for different workareas or different types of content. For example, Andre might want to make all CGIs in his workarea on `branch1` (subject to no constraints whatsoever on the arguments these CGIs or may not take) be served by `test1.example.com:1234`. This would let Andre test different Web server configurations for his CGIs on `branch1` without disturbing anyone else.

To configure TeamSite to use different Web servers, edit the [iwproxy\_preconnect\_remap] section of `iw.cfg`:

```
[iwproxy_preconnect_remap]
_regex=source_regex=dest_ex
```

where `source_regex` is a case-insensitive regular expression describing the area and files to be served by the other Web server, and `dest_ex` is an expression describing the area and files on the other Web server. This expression must include the port number.

For this to work properly, the other Web server must have the appropriate NFS TeamSite directory mounts and privileges. The Web server alias used by httpd on port 1234 of `test1.example.com` must be configured with the TeamSite alias as well (`/iw-mount/`).

The following example would allow Andre to test all CGIs in his workarea on `test1.example.com`, as described above:

```
[iwproxy_preconnect_remap]
_regex=/iw-mount/default/main/branch1/WORKAREA/andre/(.*).cgi
(\?.*)?=$http://test1.example.com:1234/iw-mount/default/main/branch1/
WORKAREA/andre/$1.cgi$2
```

## Configuring External Remappings

The TeamSite proxy server allows you to define mappings to directories outside of the TeamSite system or on different computers altogether. You can define these mappings through either of the following ways:

- `[iwproxy_preconnect_remap]`
- `[iwproxy_external_remap]`

If you use `[iwproxy_preconnect_remap]`, these mappings will follow normal `[iwproxy_preconnect_remap]` precedence rules. However, `[iwproxy_external_remap]` mappings apply *only* if no other remapping rule has been applied.

### **[iwproxy\_preconnect\_remap]**

To configure TeamSite to redirect workarea views to external Web servers, edit the `[iwproxy_preconnect_remap]` section of `iw.cfg`:

```
[iwproxy_preconnect_remap]
_regex=source_regex=dest_ex
```

where `source_regex` is a case-insensitive regular expression describing the area to be mapped from, and `dest_ex` is an expression describing the area to be mapped to. These areas are most commonly workareas or staging areas, but you can map to and from any workarea, staging area, or edition.

For example:

```
_regex=(.*)/branch1/WORKAREA/[^\/]+/logos/(.*)=$http://corporateidserver
.example.com/logos/$2
```

will send all requests for files in the `/logos` directory in all workareas on `branch1` to another server, `corporateidserver.example.com`.

### **[iwproxy\_external\_remap]**

You can also use `[iwproxy_external_remap]` rules for external remappings. This usage is being phased out; use `[iwproxy_preconnect_remap]` whenever possible.

For example, if all your corporate logo files reside on a separate server, you can use `[iwproxy_external_remap]` to create a mapping to the directory where they reside:

```
[iwproxy_external_remap]
/logos/=http://corporateidserver.example.com/logos/
```

This remapping sends all requests for `/logos/` to a directory on another server, `corporateidserver.example.com/logos/`. You can also create associations using case-insensitive regular expression mapping.

The `[iwproxy_external_remap]` section is read after the `[iwproxy_remap]` section, and it will only be applied if none of the `[iwproxy_remap]` rules were invoked. For example, if you create a mapping for `/logos/` in one of the `[branchrewrite]` sections, all requests on that branch for files in the `/logos/` directory will use that mapping *instead of* the external mapping. Requests on other branches will still be sent to the `corporateidserver`.

## **Host Header Remappings**

If your Web server manipulates “Host:” headers (for example, virtual domains), you can configure TeamSite to have the same behavior. To configure “Host:” header remapping, edit the `[iwproxy_hostheader_remap]` section of `iw.cfg`.

```
[iwproxy_hostheader_remap]
_regex=source_regex=dest_ex
```

where `source_regex` is a case-insensitive regular expression describing the area to be mapped from, and `dest_ex` is an expression describing the new “Host:” header. For example:

```
_regex=^/iw-mount/default/main/branch1/WORKAREA/.*=example.com:1234
```

will change the “Host:” header that the origin server gets from the TeamSite proxy server to read:

`Host: example.com:1234`

whenever content in a workarea on `branch1` is accessed.

## Configuring SSI Remapping

The TeamSite Web server plug-in supports the ability to both remap and virtualize SSI requests. To enable SSI request virtualization, you must install the necessary redirector module (`iwproxy_nsapi.solaris.so`) in addition to the Web server plug-in.

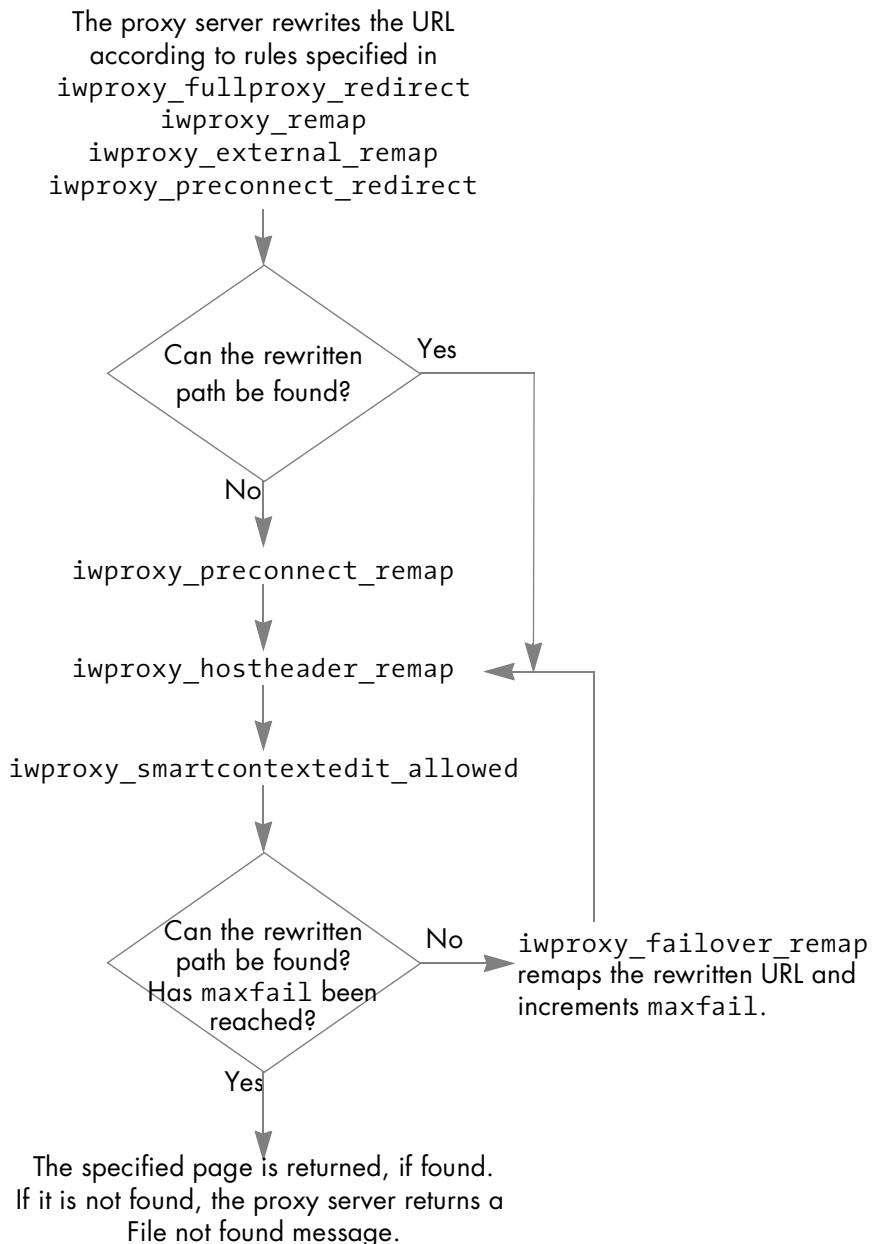
After installing the necessary redirector module as described on page 57, you can configure TeamSite to remap SSI requests by adding or modifying the `[iwproxy_plugin_remap]` section of `iw.cfg`. In the following example, any SSI request containing the string `/forms/` is mapped to `/iw-mount/default/main/Branch2/STAGING/forms` instead of being referred to the root of the user’s workarea:

```
[iwproxy_plugin_remap]
_regex=(.*)/forms/(.*)=/iw-mount/default/main/Branch2/STAGING/forms/$2
```

If you want to debug regular expressions, set the value for `_debug` in the `[iwproxy_plugin_remap]` section to `true`. On NES and Apache, debugging information is stored in the Web server error log file. This log file can grow extremely large over time.

## Configuring Proxy Failover

If a requested page does not exist, the `[iwproxy_failover_remap]` section of `iw.cfg` can be used to specify an alternate location. This section allows you to specify both alternate locations and the number of times to process an URL in an attempt to find a valid location. The figure below illustrates the process by which proxy failover remaps URLs.



The [iwproxy\_failover\_remap] section has the following structure:

```
[iwproxy_failover_remap]
_maxfail="#"
_regex=source_regex=dest_ex
_regex=source_regex=dest_ex
```

To specify the number of times to try to remap a URL, edit the \_maxfail line of the [iwproxy\_failover\_remap] section of `iw.cfg`. The default value of this line is `_maxfail=0`, which turns off proxy failover. Note that proxy failover is seldom needed because files are almost always in locations that can be specified via static, case-insensitive regular expressions during configuration. If you need to enable proxy failover, it is recommended that you do not set `_maxfail` to more than 1 or 2 due to the impact on system performance.

To specify expressions to remap, add `_regex` lines to [iwproxy\_failover\_remap]. These lines specify an incoming pattern to match, and an expression that they should be mapped to. The proxy server will take the first match it finds, remap it as specified, then try to locate the page. If it cannot find the new location, it will try to match the remapped expression to a regular expression specified in [iwproxy\_failover\_remap]. This process will continue until a match is found or the number of iterations specified by the `_maxfail` line is reached.

`_regex` lines in the [iwproxy\_failover\_remap] section follow the same syntax as `_regex` lines specified in the [iwproxy\_preconnect\_remap] section of `iw.cfg`, where `source_regex` is a case-insensitive regular expression describing the area to be mapped from, and `dest_ex` is an expression describing the area to be mapped to. For examples of `_regex` syntax, see “Resolving Relative and Absolute Paths” on page 178.

## Debugging Your Proxy Server Configuration

If your proxy server does not seem to be configured correctly, use the `iwproxy` CLT’s debug option to list all the translations being made by the proxy server:

```
iwproxy [-d|-x]
```

- |    |                                                                  |
|----|------------------------------------------------------------------|
| -d | Debug mode (outputs client & server headers)                     |
| -x | Extended (verbose) debug mode (outputs client body text as well) |

`iwproxy` will return debug output which you can redirect to a file. Note that `iwproxy`'s debug mode is single-threaded; it therefore slows the TeamSite server down tremendously. Use the debug mode for diagnostic purposes *only*.

One common source of proxy configuration problems is the inclusion of any character or blank space past the end of a branch name in any line in any [`iwproxy*`] section in `iw.cfg`. For example, the following line in the [`iwproxy_remap`] section is illegal because it contains blank spaces and characters after the branch name:

```
[iwproxy-remap]
tag_engspecs=/main/engspecs #This is the engineering spec site
```

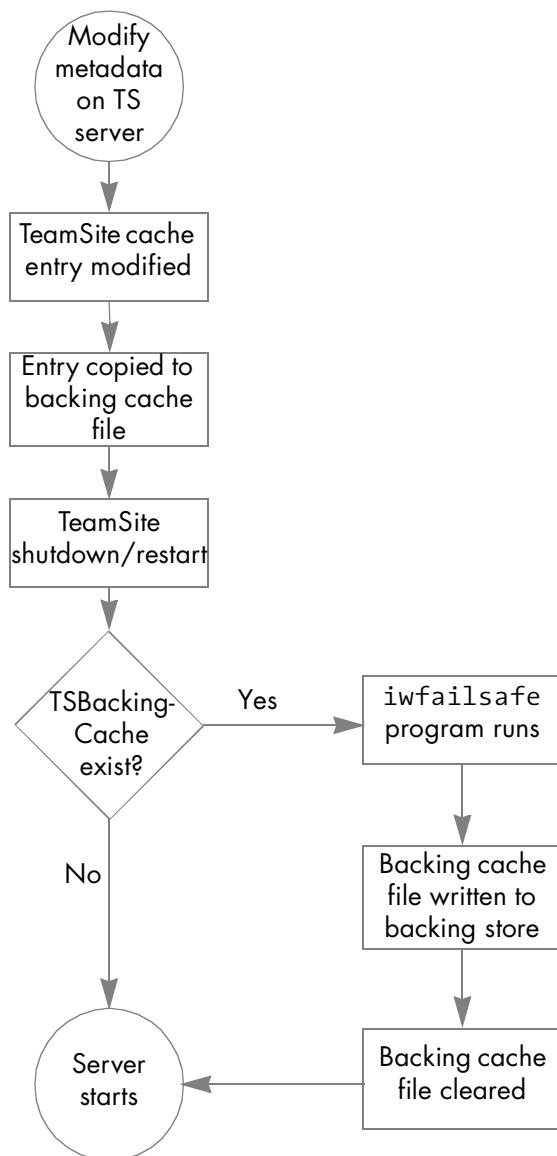
**Note:** `iwproxy` needs to run as root.

## TeamSite Embedded Failsafe

The TeamSite Failsafe functionality has been automated to improve the ability to protect your assets against unexpected server outages. Unlike previous versions of TeamSite, there is no need to modify your `iw.cfg` file to benefit from what is now known as *Embedded Failsafe*.

Embedded Failsafe improves reliability by automatically copying TeamSite cache entries to a temporary disk backup file. If the TeamSite server terminates abnormally, these cache entry copies are accessed automatically to restore the backing store when you restart the TeamSite server. This feature significantly reduces the likelihood of metadata inconsistencies caused by abnormal server termination.

The following flowchart shows the processes involved in both normal and abnormal TeamSite shutdowns.


*TeamSite Failsafe Process Flow*

## Chapter 6

# Configuring Metadata Capture and Search

---

TeamSite metadata capture lets end users add metadata information to files. After the metadata is deployed to a database via DataDeploy, end users can use TeamSite metadata search to query the database and locate files having specific metadata characteristics.

You must configure TeamSite to enable metadata capture or search; they do not appear by default in the TeamSite GUI. Configuration involves editing a set of configuration files to specify the appearance and behavior of the metadata forms, and then editing the main TeamSite configuration file (`iw.cfg`) to add metadata capture or search to a TeamSite GUI menu. After configuration is complete, end users enter information in either a metadata entry form or a metadata search form. Following data entry, the forms are processed by the metadata capture or metadata search subsystem residing on the TeamSite server. Metadata capture and search exist as separate entities, each accessed via its own TeamSite GUI menu item. Metadata capture can exist without metadata search. However, metadata search requires that you also configure metadata capture.

The rest of this chapter describes how the metadata capture and search subsystems work, and how to configure them. For details about using metadata capture and metadata search, see the *TeamSite User's Guide*.

## Metadata Capture

The following sections describe:

- A metadata capture overview.
- The main components that make up metadata capture.
- How to configure metadata capture.

## Overview

Metadata capture is a file-specific feature. That is, you must explicitly select the file(s) on which you intend to set metadata. You cannot globally set metadata for an entire area or branch. For example, to set metadata on all files in a workarea, you must select each file in that workarea (by choosing **Select All**, or by clicking the checkbox next to each file, etc.) and then initiate a metadata capture session. See the *TeamSite User's Guide* for more information.

Metadata capture can be initiated in one of two ways:

- Through a job as part of a <cgitask> element (see “Initiating Metadata Capture from a Job Specification File” on page 222), or
- From a menu item in the TeamSite GUI. A menu item for metadata capture is not on a TeamSite menu by default; you must add it as described later in this chapter.

## Components

No matter how metadata capture is initiated, it relies on four main components:

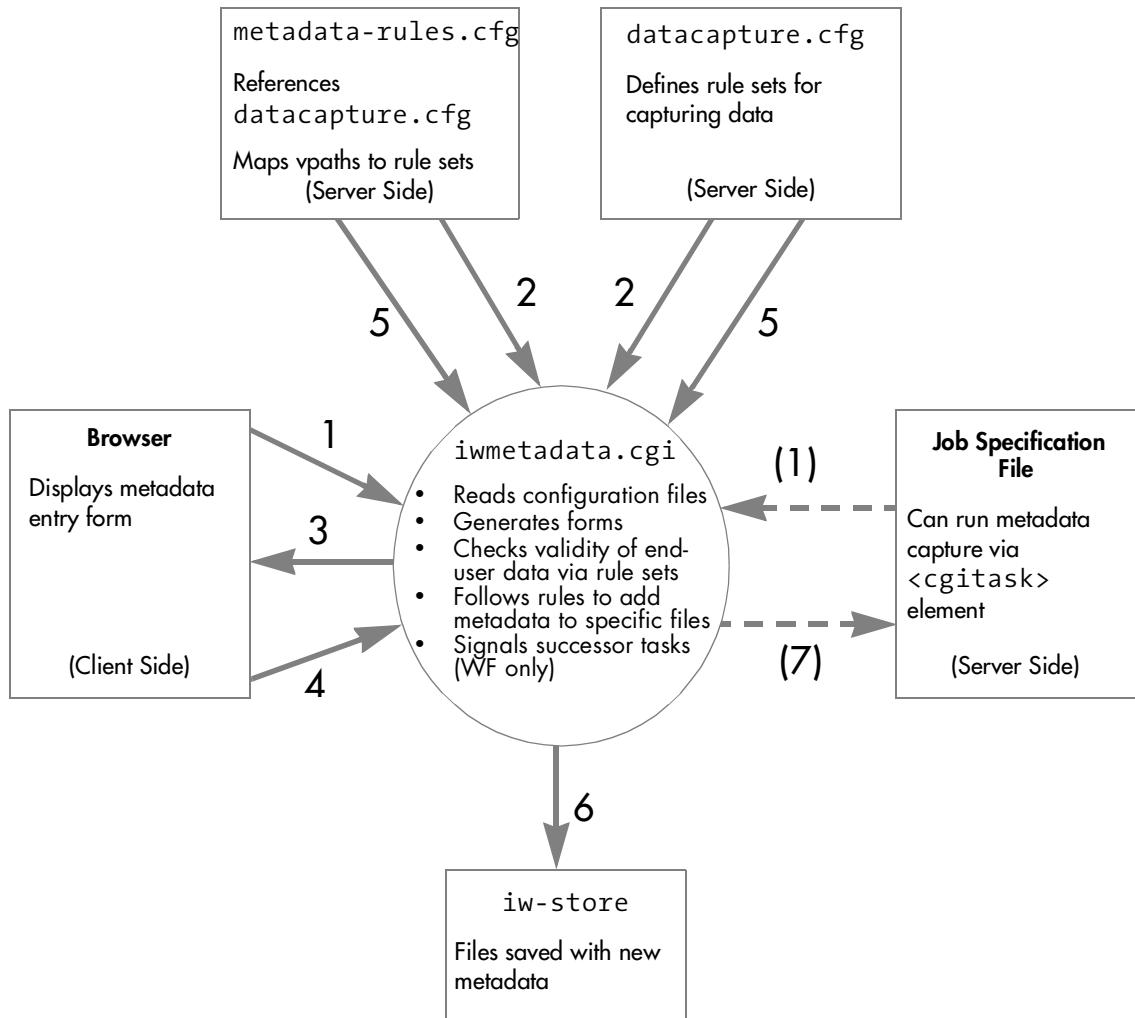
- The `iw-home/local/config/metadata-rules.cfg` configuration file, which maps vpaths to the data capture rules defined in `datacapture.cfg`.
- The `iw-home/local/config/datacapture.cfg` configuration file, which defines rule sets for capturing data.
- The metadata capture CGI `iwmetadata.cgi`, which interprets data from end users and rules in `datacapture.cfg` and `metadata-rules.cfg`, produces browser graphics and prompts, and acts as an interface with workflow configuration files (if metadata capture is running as part of a job).
- A browser interface for end-user input.

Two configuration files (`metadata-rules.cfg` and `datacapture.cfg`) allow you to configure the following on a per-user or per-vpath basis:

- The metadata item name that is displayed in the metadata entry form.
- The interface through which an end user enters input (for example, a checkbox or a data field).

- The type of data that is acceptable or unacceptable in any given field.
- Whether input is required for any given field.

The following diagram shows how these components work together. Sections following the diagram explain each diagram step and component in detail.



Metadata Capture Overview

## Diagram Key

1. The metadata CGI receives a list containing the names of the files that will have metadata added to them. The list can come from an instantiated job (if metadata capture is initiated from a job) or from the browser (if initiated from the TeamSite GUI).
2. The metadata CGI reads both configuration files (`metadata-rules.cfg` and `datacapture.cfg`) to determine what information it should display in the metadata entry form. It makes this determination on a per-file basis, so that the entry form can contain different prompts and actions for different files.
3. The metadata CGI displays the metadata entry form on the client system via the GUI.
4. An end user fills in data and submits the entry form back to the metadata CGI.
5. The metadata CGI consults the rules in both configuration files to verify the validity of the data entered by the end user. If the data does not meet all necessary criteria, notification is sent to the end user so that data can be re-entered.
6. If the data meets all necessary criteria, the metadata CGI adds the new metadata (in the form of TeamSite extended attributes) to the specified files. The metadata CGI interfaces directly with the backing store to update the files with the new metadata.
7. If metadata capture was initiated from a job, the metadata CGI notifies the workflow subsystem, which starts successor task 0 (zero) as defined in the job specification file.

## Configuring Metadata Capture

You must perform three main activities to configure metadata capture:

1. Create a `metadata-rules.cfg` file in `iw-home/local/config` for your site.
2. Create a `datacapture.cfg` file in `iw-home/local/config` for your site.
3. Add a **Set Metadata** item to the TeamSite GUI so that end users can access metadata capture.

The following sections describe these steps in detail.

## Configuring metadata-rules.cfg

The `metadata-rules.cfg` file maps vpaths to data capture rules that are defined in `datacapture.cfg`. The `metadata-rules.cfg` file consists of a series of `<cond>` (conditional) elements. A `<cond>` element can contain `<rule>` elements and other `<cond>` elements. Each vpath is run through `metadata-rules.cfg`, resulting in a one-to-many mapping from vpaths to named rules. Whenever a list of `<cond>` elements is found, the first to match the current vpath takes effect, and the rest of the elements in the list are discarded.

When you set up `iw-home/local/config/metadata-rules.cfg` for your site, it is recommended that you copy and edit the example file provided with TeamSite (`iw-home/local/config/metadata-rules.cfg.example`). Use the following DTD and annotated examples as references for your own site-specific configuration.

### ***DTD: metadata-rules.cfg***

The `metadata-rules.cfg` file uses the following DTD:

```
<!ELEMENT metadata-rules (cond)*>
<!ELEMENT cond (cond|rule)*>
<!ATTLIST cond
 vpath-regex CDATA #REQUIRED
 >
<!ELEMENT rule EMPTY>
 <!ATTLIST rule
 name CDATA #REQUIRED
 >
```

### ***Sample metadata-rules.cfg File 1***

The following `metadata-rules.cfg` file is distributed with TeamSite as `iw-home/local/config/metadata-rules.cfg.example`.

```
<?xml version="1.0" encoding="UTF-8" ?> ← International Encoding 1
<metadata-rules>
 <cond vpath-regex="."> ← Vpath Identifier 2
 <rule name="Default Rule" /> ← Rule Identifier 3
 </cond>
</metadata-rules>
```

### *Sample metadata-rules.cfg File 1 Notes*

- 1. International Encoding:** UTF-8 is an encoding of Unicode, a standard for encoding the character sets of international languages. All web assets should specify their encoding as UTF-8. For details about web asset encoding, see Appendix D, “Internationalization”.
- 2. Vpath Identifier:** Names the vpath (in this case all directories) to which the rule(s) named in the following subelement(s) will be applied.
- 3. Rule Identifier:** Names the rule that applies to the preceding vpath. The rule itself is defined in the `<ruleset>` element in `iw-home/local/config/datacapture.cfg`. In this example, the `Default Rule` rule defined in `datacapture.cfg` will always apply to all directories.

## *Sample metadata-rules.cfg File 2*

The following metadata-rules.cfg file illustrates a more sophisticated example:

```

<metadata-rules>
 <cond vpath-regex="^/default/main/syndication">
 <rule name="Default" /> ← Vpath Identifier 1
 <rule name="Syndication" /> ← Rule Identifiers 2
 <cond vpath-regex=".pdf$"> ← Vpath Identifier 3
 <rule name="PDF Files" /> ← Rule Identifier 4
 </cond>
 <cond vpath-regex=".doc$"> ← Vpath Identifier 5
 <rule name="MS Word Files" /> ← Rule Identifier 6
 </cond>
 </cond>

 <cond vpath-regex="^/default/main/www"> ← Vpath Identifier 7
 <rule name="Default" /> ← Rule Identifiers 8
 <rule name="Web Content" /> ←
 <cond vpath-regex=".html$"> ← Vpath Identifier 9
 <rule name="HTML Files" /> ← Rule Identifier 10
 <cond vpath-regex="/pr/"> ← Vpath Identifier 11
 <rule name="PR" /> ← Rule Identifier 12
 </cond>
 <cond vpath-regex="/corp/"> ← Vpath Identifier 13
 <rule name="Corporate" /> ← Rule Identifier 14
 </cond>
 </cond>
 </cond>
</metadata-rules>
```

## *Sample metadata-rules.cfg File 2 Notes*

- 1. Vpath Identifier:** Files on the /main/syndication branch will always receive the rules named in the following subelements.
- 2. Rule Identifiers:** The Default and Syndication rules defined in datacapture.cfg will always apply to the /main/syndication branch.

3. **Vpath Identifier:** Files ending in .pdf on the /main/syndication branch will receive rules in addition to those defined by Default and Syndication.
4. **Rule Identifier:** The PDF Files rule defined in datacapture.cfg will apply to files ending in .pdf on the /main/syndication branch.
5. **Vpath Identifier:** Files ending in .doc on the /main/syndication branch will receive rules in addition to those defined by Default and Syndication.
6. **Rule Identifier:** The MS Word Files rule defined in datacapture.cfg will apply to files ending in .doc on the /main/syndication branch.
7. **Vpath Identifier:** The /main/www branch will always receive the rules named in the following subelements.
8. **Rule Identifiers:** The Default and Web Content rules defined in datacapture.cfg will apply to the /main/www branch.
9. **Vpath Identifier:** Files ending in .html on the /main/www branch will receive rules in addition to those defined by Default and Web Content.
10. **Rule Identifier:** The HTML Files rule defined in datacapture.cfg will apply to files ending in .html on the /main/www branch.
11. **Vpath Identifier:** Files ending in .html in the pr directory on the /main/www branch will receive rules in addition to those defined by Default and Web Content.
12. **Rule Identifier:** The PR rule defined in datacapture.cfg will apply to files ending in .html in the pr directory on the /main/www branch.
13. **Vpath Identifier:** Files ending in .html in the corp directory on the /main/www branch will receive rules in addition to those defined by Default and Web Content.
14. **Rule Identifier:** The Corporate rule defined in datacapture.cfg will apply to files ending in .html in the corp directory on the /main/www branch.

## Configuring `datacapture.cfg`

The `datacapture.cfg` file defines rule sets for capturing data. Rules are referred to by name in `metadata-rules.cfg` (see “Configuring `metadata-rules.cfg`” on page 201).

Rules contain *items*, where each item is a single set of data that is to be captured from the end-user. An item consists of one or more *instances*. Each instance encapsulates how to capture the data for the item, and each instance defines an ACL that determines which (if any) instance a particular user is allowed to use to enter the data.

The metadata capture form is a data capture template (DCT) that is configured specifically for metadata capture. The DCT subsystem that generates the metadata capture form is the same subsystem that generates DCTs for TeamSite Templating. A major difference between the two implementations is the location of the `datacapture.cfg` file. TeamSite Templating relies on multiple `datacapture.cfg` files (one for each data type), while metadata capture relies on a single `datacapture.cfg` file (in `iw-home/local/config`).

See “Setting Up Data Capture Templates” in the *TeamSite Templating Developer’s Guide* for a complete explanation of `datacapture.cfg` files, including annotated examples and explanations of elements and attributes. Note that even though the examples in the *TeamSite Templating Developer’s Guide* are specific to TeamSite Templating, they are useful as reference points for setting up `datacapture.cfg` for metadata capture.

An example `datacapture.cfg` file configured specifically for metadata capture is also included with TeamSite (refer to `iw-home/local/config/datacapture.cfg.example`). An annotated explanation of that file is shown in “Sample `datacapture.cfg` File 1” on page 208.

When you set up `iw-home/local/config/datacapture.cfg` for your site, it is recommended that you copy and edit the `datacapture.cfg.example`, using the following DTD and annotated examples as reference points for your own site-specific configuration.

### ***DTD: datacapture.cfg***

The datacapture.cfg file uses the following DTD. This DTD is also available online in `iw-home/local/config`.

```
<!ELEMENT data-capture-requirements (ruleset)*>
<!ATTLIST data-capture-requirements
 name CDATA #REQUIRED
 type(metadata|content|workflow) #REQUIRED
>

<!ELEMENT ruleset (item)*>
<!ATTLIST ruleset
 name CDATA #REQUIRED
>

<!ELEMENT item (database?,(%instance;)*)>
<!ATTLIST item
 name CDATA #REQUIRED
 description CDATA #IMPLIED
>

<!ENTITY % instance "(checkbox|radio|text|textarea|select|replicant)" >

<!ELEMENT checkbox(allowed|option)*>
<!ATTLIST checkbox
 required (t|f)"f"
 delimiter CDATA", "
>
<!ELEMENT radio (allowed|option)*>
<!ATTLIST radio
 required (t|f) "f"
>
<!ELEMENT text (allowed)*>
<!ATTLIST text
 required (t|f) "f"
 maxlen NUM
 size NUM
 validation-regex CDATA -- regex(5) for validating this element -
>
```

```
<!ELEMENT textarea (allowed)*>
<!ATTLIST textarea
 required (t|f) "f"
 rows NUM
 cols NUM
 validation-regex CDATA -- regex(5) for validating this element --
 >
<!ELEMENT select (allowed|optgroup|option)*>
<!ATTLIST select
 required (t|f) "f"
 size NUM
 multiple (t|f) "f"
 delimiter CDATA ", " -- for multiple=t only --
 >
<!ELEMENT replicant (allowed|item)*>
<!ATTLIST replicant
 min NUM
 max NUM
 default NUM
 >

<!ELEMENT optgroup (optgroup*, option*)+>
<!ATTLIST optgroup
 label CDATA #REQUIRED
 >

<!ELEMENT option EMPTY>
<!ATTLIST option
 selected (t|f) "f"
 value CDATA #IMPLIED
 label CDATA #REQUIRED
 >

<!ELEMENT database EMPTY >
<!ATTLIST database
 deploy-column(t|f) "t"
 searchable (t|f) "t"
 data-type CDATA "VARCHAR(255)"
 data-format CDATA #IMPLIED
 >
```

```
<!ELEMENT allowed (cred|and|or|not)>

<!ELEMENT cred EMPTY>
<!ATTLIST cred
 role CDATA #IMPLIED
 user CDATA #IMPLIED
 >

<!ELEMENT and (cred|and|or|not)+>

<!ELEMENT or (cred|and|or|not)+>

<!ELEMENT not (cred|and|or|not)>
```

### *Sample datacapture.cfg File 1*

The following `datacapture.cfg` file is distributed with TeamSite as `iw-home/local/config/datacapture.cfg.example`. See the section immediately following the file for an explanation of the numbered callouts. See the *TeamSite Templating Developer's Guide* for a complete explanation of `datacapture.cfg` files.

```

<?xml version="1.0" encoding="UTF-8" ?> ← International Encoding 1

<!-- A <data-capture-requirements> element with type="metadata"
can contain multiple <ruleset> elements.

Note: The <database> elements have no effect on the metadata
capture process. These optional elements are used to help integrate
metadata capture with Data Deploy. Data Deploy configuration
files can be automatically generated from datacapture.cfg files.
The <database> tags ensure the database tables are built using
the appropriate datatype.

-->

<data-capture-requirements type="metadata"> ← Metadata Identifier 2
 <ruleset name="Default Rule"> ← Rule Identifier 3

 <description>
 This rule applies to all files on all branches.
 </description>

 <item name="Title">
 <database searchable="t" data-type="VARCHAR(60)" /> ← database Element 4
 <text required="t" maxlength="60" /> ← Instance (text) 5
 </item>

 <item name="Description">
 <database data-type="VARCHAR(100)" />
 <text required="t" maxlength="100" />
 </item>

 <item name="Type">
 <database data-type="VARCHAR(30)" />
 <select>
 <option label="White Paper" value="white_paper" />
 <option label="Datasheet" value="datasheet" />
 <option label="Press Release" value="press_release" />
 <option label="Architecture Overview" value="architecture" />
 <option label="Futures Overview" value="futures" />
 <option label="Program Material" value="program_material" />
 </select>
 </item>

```

```
<item name="Category">
 <database data-type="VARCHAR(40)" />
 <!-- To use the example callout,
 1. Comment out this select element.
 2. Uncomment the text element.
 -->
 <select>
 <option label="Internet - Financial" value="financial_internet"/>
 <option label="Internet - Manufacturing" value="manufacturing_in
 ternet" />
 <option label="Internet - Services" value="services_internet" />
 <option label="Extranet - Tier 1" value="tier_1_extranet" />
 <option label="Extranet - Tier 2" value="tier_2_extranet" />
 <option label="Extranet - Tier 3" value="tier_3_extranet" />
 </select>
 <!--
 <text>
 <callout type="cgi"
 label="Query for Categories"
 url="/iw-bin/iw_cgi_wrapper.cgi/example_datacapture_callout.i
 pl/metadata-category-options.txt" />
 </text>
 -->
</item>

<item name="Languages">
 <database data-type="VARCHAR(20)" />
 <checkbox>
 <option label="English" value="English" />
 <option label="German" value="German" />
 <option label="French" value="French" />
 <option label="Japanese" value="Japanese" />
 <option label="Chinese" value="Chinese" />
 </checkbox>
</item>

<item name="Source">
 <database data-type="VARCHAR(50)" />
 <text maxlength="50" />
</item>
```

```
<item name="Launch Date">
 <database data-type="DATE" data-format="yyyy-MM-dd" />
 <text required="t" maxlength="10" validation-regex="^[0-9][0-9]
 [0-9][0-9]-[0-1][0-9]-[0-3][0-9]$" />
</item>
<item name="Expiration Date">
 <database data-type="DATE" data-format="yyyy-MM-dd" />
 <text maxlength="10" validation-regex="^[0-9][0-9][0-9][0-9]-[0-1][
 0-9]-[0-3][0-9]$" />
</item>
<item name="Keywords">
 <database data-type="VARCHAR(100)" />
 <text maxlength="100" />
</item>
</ruleset>
</data-capture-requirements>
```

The XML code defines three items: Launch Date, Expiration Date, and Keywords. The Launch Date and Expiration Date items both have a database data-type of "DATE". Arrows point from these two entries to the text "DATE datatype" with a superscript 6. The Launch Date item has a validation-regex of "^[0-9][0-9][0-9][0-9]-[0-1][0-9]-[0-3][0-9]\$". An arrow points from this regex to the text "validation-regex" with a superscript 7.

## *Sample datacapture.cfg File 1 Notes*

The following information is specific to the example file shown in the preceding section. For more detailed information about `datacapture.cfg` files, see the *TeamSite Templating Developer's Guide*.

- 1. International Encoding:** UTF-8 is an encoding of Unicode, a standard for encoding the character sets of international languages. All web assets should specify their encoding as UTF-8. For details about web asset encoding, see Appendix D, "Internationalization".
- 2. Metadata Identifier:** When configuring `datacapture.cfg` for metadata capture, you must specify "type=metadata" in the `<data-capture-requirements>` element as shown here.
- 3. Rule Identifier:** The `<ruleset>` element contains all of the items that make up the rule set that defines the appearance and behavior of the data capture form. A `datacapture.cfg` file that is configured for metadata capture can contain any number of `<ruleset>` elements (as opposed to TeamSite Templating `datacapture.cfg` files, which can contain just one `<ruleset>` element). This example file happens to contain just one `<ruleset>` element; it could contain more if necessary. The rule defined here is named Default Rule, and is referenced by the `metadata-rules.cfg` file shown page 202. The name attribute is required and its value appears in the TeamSite GUI as the name of the data capture form. More than one form can appear on a single page. Optional subelements `<label>`, `<description>`, `<item>`, and `<itemref>`. The `<label>` subelement is used to provide a label on the data capture form. The `<itemref>` subelement requires the name attribute and is used as a stand-in for the `<item>` subelement in a `<symbol-table>` element.
- 4. database Element:** The optional `<database>` element facilitates the use of the appropriate data type in DataDeploy and is used only for generation of the `mdc_dd.cfg` file. It does not control any aspects of the metadata capture or search forms. The `<data-type>` and `<searchable>` information in the `<database>` element are passed on to `mdc_dd.cfg`, which in turn uses that information to control how metadata is deployed to a database via DAS. The `<database>` element has four attributes. Because attribute order in XML documents is important; these attributes should occur in the order they are listed:

- **deploy-column** can be either "t" (default) or "f" and allows you to set whether or not data entered for the item is deployed to a database column.
- **searchable** can be either "t" (default) or "f" and allows you to set whether or not users can search against this item.
- **data-type** is required and is any JDBC database type. If you do not set the **data-type** attribute, a default datatype of VARCHAR (255) is set in `mdc_dd.cfg`.
- **data-format** describes the format if **date** or **time** is specified for the **data-type** attribute (see callout 6). If a value for **data-format** is specified, the instance should contain a validation regex to force a valid entry in the field (see callout 7).

In this example, **deploy-column** would be the first attribute if it were set. It is not, so all input for this item will be deployed to a database column. Next, the **searchable attribute** is specified as "t"; however, because this is the default value for the attribute, **searchable** need not be included here. Following **searchable** is **data-type**, here specifying the input to be stored as a string. If date or time is set as the value for the **data-type**, a **data-format** attribute should end the element.

**5. Instance (text):** The optional `<text>` element controls the length of text entry fields in metadata capture and search forms. It also controls whether an end user is required to enter text in a field. If the datatype is **date** or **time** and a format has been specified, it is best to include a **validation-regex** to force users to input data in the correct format (see callout 7). In this example the user is required enter a string of between one and 60 characters in the text field. The data entered for this item is stored in the database as VARCHAR and is searchable.

**6. DATE datatype:** If the datatype is set to **date** or **time**, it is recommended you specify a **data-format** and include a validation. Because there are many formats for date and time, specifying a format forces the user to enter data in that format and reduces the chance of user error. The value for **data-format** can be any valid Java format for a date or time.

**7. validation-regex:** The user can be forced to enter a date or time in the format you specify by including a validation regex. The value for the **validation-regex** attribute must match the format specified in for **data-format**. The regex in this example specifies the range of digits that can be entered for `yyyy-mm-dd` and that dashes must separate year, month and day. The following table shows validation regex examples for several supported datatypes. The `<database>` and `<text>` elements shown in the table are

subelements of the <item> element. Some regex lines are wrapped due to formatting constraints. You should enter them all on one line in your configuration file.

Datatype	Notes	Example
DATE	If data-type is DATE, the data-format must be a format string that is valid for the Java simple date format class. Formats do not have to be year-month-day, any valid format will work.	<pre>&lt;database data-type="DATE" data-format="yyyy-mm-dd" /&gt; &lt;text maxlength="10"       validation-regex="^[0-9][0-9][0-9][0-9]-[0- 1][0-9]-[0-3][0-9]\$" /&gt;</pre>
INT	Allows any integer up to 7 digits. This example assumes that you want to store data as integers, not dollars and cents.	<pre>&lt;database data-type="INT" /&gt; &lt;text maxlength="7"       validation-regex="^[0-9]\{0,\} \$" /&gt;</pre>
REAL	Allows any decimal up to 8 digits (including decimal). The regex allows 0 or more digits, followed by a decimal point, followed by zero or more digits.	<pre>&lt;database data-type="REAL" /&gt; &lt;text required="t" maxlength="8"       validation-regex="^[0-9]\{0,\}\.[0-9]\{0,\} \$" /&gt;</pre>

### ***Sample datacapture.cfg File 2***

The following `datacapture.cfg` file is written to work with the file shown earlier in “Sample metadata-rules.cfg File 2” on page 203.

```

<!-- This config file defines rulesets for capturing data.
Rules are referred to by name in other config files, such as
metadata-rules.cfg. Rules contain "items"; one item is a single
(set of) data that is to be captured from the end user.
An item consists of one or more "instances". Each instance
encapsulates how to capture the data for the item, and each
instance defines an ACL that determines which (if any)
instance a particular user is allowed to use to enter the
data. Instances are text, textarea, radio, checkbox, select,
and replicant. (Others are coming.)
Replicants are very special kinds of instances; they are
repeatable. Replicants contain _items_ instead of just an ACL
like the other types of instances.

-->
<data-capture-requirements type="metadata">
 <ruleset name="Default">
 <item name="Author">
 <database data-type="VARCHAR(12)" />
 <!-- This item is represented by a text box. -->
 <text size="12" required="t" />
 <!-- no ACL means open access for everyone -->
 </item>
 </ruleset>
 <ruleset name="Syndication">
 <item name="Category">
 <database data-type="VARCHAR(10)" />
 <!-- This item is represented by a series of four
 checkboxes. -->

 <checkbox required="t" delimiter="/"> ← Instance (checkbox) with delimiter1

 <!-- We want the TeamSite extended attribute
 to use "/" as the value delimiter when
 concatenating all the selected values,
 e.g., "Partners/Customers." -->

 <option label="Partners" />
 <option label="Suppliers" />
 <option label="Customers" />
 <option label="Internal" />

```

```

<ruleset name="PDF Files">
 <item name="All Keywords">
 <!-- All nested <item> elements must be "type compatible"
 if the fields are going to be deployed to a database. -->
 <database data-type="VARCHAR(20)" searchable="f" />
 <!-- Because any number of keywords may apply to a
 single file, we use a replicant instance for
 the "keywords" item. -->
 <replicant default="3" min="1" max="12"> ← Instance (replicant) 2
 <!-- We allow from 1 to 12 keywords. -->
 <!-- This replicant instance contains just one item,
 which has two instances. -->
 <!-- When there are multiple instances, the first
 instance whose ACL allows the current user
 will be the instance used for that user. -->
 <item name="Keyword">
 <text size="20" required="t">
 <!-- This ACL allows "joe" and masters
 to type anything she wants. -->
 <allowed> ← Access Control Limiter (ACL) 3
 <or>
 <cred user="joe" /> ← Access Identifier 4
 <cred role="master" />
 </or>
 </allowed>
 </text>
 <select required="t">
 <!-- Everyone but joe has to choose from
 pre-determined choices. -->
 <allowed>
 <not>
 <cred user="joe" />
 </not>
 </allowed>
 <option label="supply chain" />
 <option label="marketing" />
 <option label="sales promotions" />
 <option label="earnings" />
 <option label="facilities" />
 <option label="eCommerce" />
 </select>
 </item>
 </replicant>
 </item>

```

```

<ruleset name="MS Word Files">
 <!-- ... -->
</ruleset>
<ruleset name="Web Content">
 <!-- ... -->
</ruleset>
<ruleset name="HTML Files">
 <!-- ... -->
</ruleset>
<ruleset name="PR">
 <item name="Go-Live Date">
 <!-- The database column must allow strings like "Today" and
 "Tomorrow", so the DATE datatype cannot be used.
 This configuration decreases the usefulness of searching
 this database column -->

 <database searchable="f" data-type="VARCHAR(10)" />

 <!-- This text instance has a regular expression that
 determines validity of user-entered data.
 In this case, the regex requires the user
 to enter "#/#/#/#". -->
 <text size="10" validation-regex="^[0-9][0-9]/[0-9][0-9]/[0-9][0-9] [0-9][0-9]$">
 <allowed>
 <or>
 <cred role="editor" />
 <cred role="admin" />
 <cred role="master" />
 </or>
 </allowed>
 </text>
 <select required="t">
 <option label="Today" />
 <option label="Tomorrow" selected="t" />
 <option label="Next Week" />
 <allowed>
 <cred role="author" />
 </allowed>
 </select>
 </item>
</ruleset>
<ruleset name="Corporate">

```

Variable Instances 5

## **Sample *datacapture.cfg* File 2 Notes**

The following information is specific to the example file shown in the preceding section. For more detailed information about *datacapture.cfg* files, see the *TeamSite Templating Developer's Guide*.

- 1. Instance (checkbox or select) with delimiter:** Specifies the delimiting character used when data from all check boxes is concatenated by the data capture subsystem. The default delimiter is a comma (,). In this example a “/” is used as the delimiter to separate the concatenated values for the checked `<option>` elements.
- 2. Instance (replicant):** Specifies a repeatable instance that can contain multiple nested items and instances. When there are multiple instances, the first instance whose ACL allows the current user to enter data will be the instance used for that user. `<replicant>` is the only instance that can contain nested items and instances. Whenever additional iterations of the instance can be displayed (that is, if the `max` threshold has not yet been reached), an **File > Add Above** and **File >Add Below** menu items are active. Whenever iterations of the instance can be removed (that is, if the `min` threshold has not yet been reached), The **File > Delete** menu item is active. If a `<replicant>` has four items, the **Add** menu item displays another set of four items in the data capture form. In this example, if the user's username is `joe` or role is `master`, three keyword text fields display; if the user is not a master or “`joe`”, then three drop-down selection boxes display. Keyword instances can be added or removed to a minimum of one and a maximum of 12 using the **Add** or **Delete** options in the **File** menu.
- 3. Access Control List (ACL):** The `<allowed>` element lets you set an ACL to specify which users can or cannot use a specific instance to enter data. If `<allowed>` is not set, the instance is visible to and can be used to input data by any user. The allowed element can have any of the following elements:
  - `<cred>` lets you name a user or role in the ACL.
  - `<and>` defines multiple users or multiple roles that can use the instance.
  - `<or>` defines users and roles that can use the instance.
  - `<not>` defines a user or role that is not allowed to use the instance. The instance does not display for users not allowed to use that instance.

In this example **Keyword** text fields display for the user with username `joe` or the role `master`; for other users only **Keyword** selection drop-down menus display.

**4. Accessor Identifier:** The `<cred>` element is a child element of `<and>`, `<not>`, or `<nor>` and enables to you identify the accessor by role and username. Note that `<cred>` requires exactly one attribute, either `role` or `user`. In this example two `<cred>` elements are combined under the parent `<and>` element so that the ACL applies to both the username `joe` and role `master`. Text fields display for users with either identification, while drop-down selection menus display for others.

**5. Variable Instances:** Within an `<item>` an instance can be made available to certain users or roles and not to others. In this example all Editors, Administrators, or Masters are offered text fields and can input variable text strings, while Authors are offered drop-down menus with predefined choices. Note that because there is one `<database>` element for each `<item>`, and that input for this item could be either dates or a character string, the datatype must be set to `VARCHAR`. It is not recommended that you create a `VARCHAR` database in which numerical data, such as dates or time, might be stored; operands useful for retrieving dates and time such as “between”, “less than”, “greater than” cannot be used to search such information.

### **Adding Metadata Capture to the TeamSite GUI**

Because metadata capture is a file-specific feature, it is recommended that end users access it via the **File** menu in the TeamSite GUI. To add a **Set Metadata** item to the TeamSite GUI’s **File** menu, add the following line to the `[iwcfg]` section of `iw.cfg`:

```
custom_menu_item_metadata="File", "Set Metadata", "iwmetadata.cgi"
"all" "width=800,height=570,scrollbars=yes,resizable=yes"
```

This line specifies the following:

- The TeamSite GUI menu (**File**) to which the item will be added.
- The name of the new item (**Set Metadata**).
- The CGI (`iwmetadata.cgi`) that will execute when the item is selected.
- Which users (`all`) can see the menu item.
- The appearance and behavior of the window in which the CGI runs.

See “Custom Menu Items” on page 134 for more information about adding and enabling custom menu items.

## Metadata Capture End Result

After you configure metadata capture, end users can access it via the TeamSite GUI to set metadata on files. The end result of a metadata capture session is the addition of TeamSite extended attributes to one or more files. For example:

**File:** /default/main/www/WORKAREA/jk/pr/BigAnnouncement.html

Name	Value
TeamSite/Metadata/Author	jk
TeamSite/Metadata/Go-Live Date	07/04/2000

**File:** /default/main/syndication/WORKAREA/bill/fall2000.pdf

Name	Value
TeamSite/Metadata/Author	bill
TeamSite/Metadata/Category	Partners/Customers/Internal
TeamSite/Metadata/Category/Partners	Y
TeamSite/Metadata/Category/Customers	Y
TeamSite/Metadata/Category/Internal	Y
TeamSite/Metadata/Keywords/0/Keyword	supply chain
TeamSite/Metadata/Keywords/1/Keyword	earnings
TeamSite/Metadata/Keywords/2/Keyword	eCommerce

These are the extended attributes that would be displayed via the TeamSite GUI from the **File > File Properties** menu. These extended attributes can now be deployed to a database via DataDeploy. the deployment can be manual, or automatic through Database Auto-Synchronization (DAS). See the *DataDeploy Administration Guide* for more information.

## Metadata Capture and TeamSite Workflow

The following sections describe key interactions between metadata capture and TeamSite workflow.

### Specifying Files in Workflow Tasks

Metadata capture includes the ability to self-filter a list of files on which to capture metadata. For example, a user task or a job task can name a set of files upon which metadata will be set. All symlinks, directories, and deleted files that are part of the file set will be filtered out and ignored. Only actual files will have metadata set.

### Initiating Metadata Capture from a Job Specification File

The following sample <cgitask> section from a job specification file shows the syntax necessary to initiate a typical metadata capture process from within a job. The task owner in this example is jk. The task in this example is associated with the area shown in areavpath. See the *TeamSite Workflow Developer's Guide* for the <cgitask> DTD and other general information.

```
<cgitask name="metadata" owner="jk">
 <description>apply metadata.</description>
 <areavpath v="/default/main/test/WORKAREA/jk" immediate ="+"/>
 <successors>
 <successorset description="set">
 <succ v="confirm" />
 </successorset>
 </successors>
 <command v="iwmetadata.cgi" />
 <activation>
 <or>
 <pred v="start" />
 </or>
 </activation>
</cgitask>
```

# Metadata Search

The following sections describe:

- A metadata search overview.
- The prerequisites for configuring metadata search.
- The main components that make up metadata search.
- How to configure metadata search.

## Overview

The metadata search subsystem uses search parameters supplied by an end user via a search form to query a database containing metadata. The end result is a list of files, displayed in the TeamSite GUI, that contain metadata tags matching the search parameters. The search form is based on configuration files also used by metadata capture and generated by DAS. This relationship ensures that the search form contains fields only for data that is already stored in the metadata database. Details about these files are presented later in this section.

Metadata search is an area-specific feature. That is, it performs a search for metadata tags on files in the entire area and all subareas from which it was executed. For example, if you execute metadata search from /default/main/www/WORKAREA/w1, all files in w1 and its subdirectories are searched. If you execute metadata search from /default/main/www/WORKAREA/w1/marketing, all files in marketing and its subdirectories are searched. You can execute metadata search from a workarea or any subdirectory within a workarea. You cannot execute it from a staging area, edition, or branch. See the *TeamSite User's Guide* for more information about metadata search usage.

## Prerequisites

It is essential that you configure the following features before configuring and running metadata search:

- Metadata capture, as described earlier in this chapter. This is required because metadata search relies on the same `datacapture.cfg` file as metadata capture. If this file is not configured correctly, metadata search will not run.

- DataDeploy's Database Auto-Synchronization (DAS) module as described in the *DataDeploy Administration Guide*. This is required because metadata search relies on the `mdc_dd.cfg` file, which is generated automatically when DAS is configured. If DAS is not configured correctly, metadata search will not run.

In addition, you must already have deployed metadata to a database via DAS before running metadata search. This is required because metadata search searches the database specified in the DAS configuration files. It does not search the TeamSite backing store or any database not specified in the DAS configuration files.

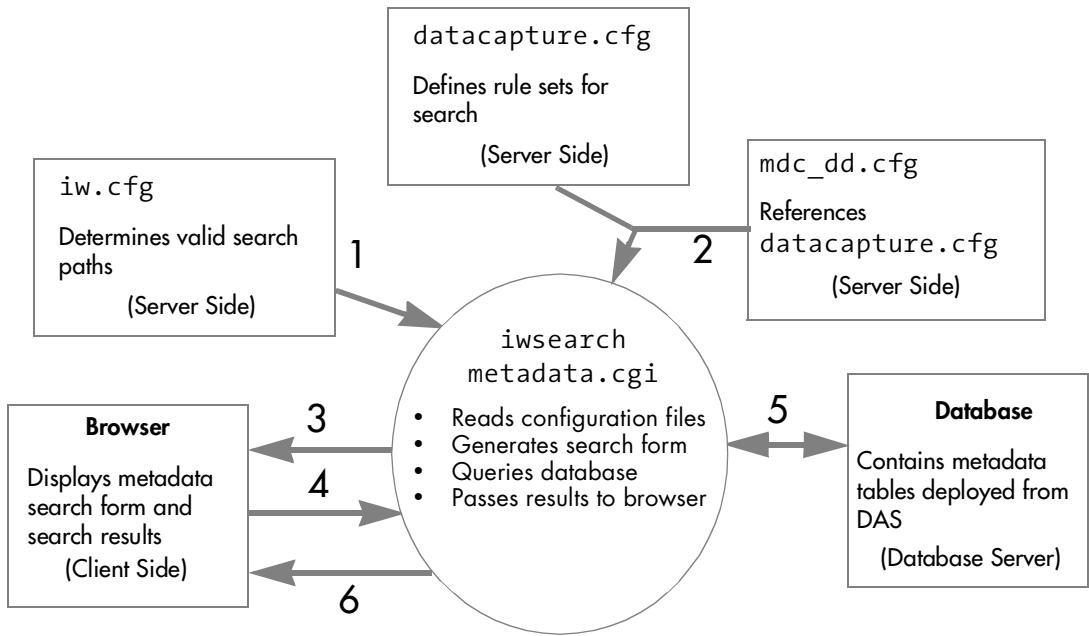
If all of these prerequisites are met, you can proceed with the metadata search configuration as described in “Configuring Metadata Search” on page 226. It is recommended that you read the following “Components” section before performing the configuration.

## Components

Metadata search relies on six main components:

- The same `iw-home/local/config/datacapture.cfg` configuration file used by the metadata capture subsystem.
- The DataDeploy configuration file `iw-home/local/config/mdc_dd.cfg`, which is generated automatically when you configure DataDeploy Database Auto-Synchronization (DAS) or when you execute the `iwsyncdb.ip1 -mdcddgen` command.
- The metadata search CGI `iwsearchmetadata.cgi`, which interprets data from end users and rules in `datacapture.cfg` and `mcd_dd.cfg`, and produces browser graphics and prompts.
- The [valid\_search\_paths] section of `/etc/iw.cfg`.
- A browser interface for end-user input.
- The database containing metadata deployed via DAS.

The following diagram shows how these components work together. Sections following the diagram explain each diagram step and component in detail.

**Diagram Key**

1. The metadata search CGI reads the `[valid_search_paths]` section of the `/etc/iw.cfg` configuration file to determine the paths where metadata search is valid. By default, all paths are considered valid search paths. For more information see “Changing Valid Search Paths” on page 228.
2. The metadata search CGI reads the `mdc_dd.cfg` and `datacapture.cfg` configuration files. The information from these files is used by the search CGI to determine what should be displayed in the metadata search form. The information provided by `mdc_dd.cfg` controls whether a metadata field is searchable, what label each field has, and which operators are valid for each field in the search form. For example, if a metadata tag residing on the database uses a data type of `CHAR`, the search form will contain operators such as `Contains`, `Does contain`, etc. for that specific field. The end user can then select one of these field-specific operators to set the search parameters for that field. However, if a metadata tag uses a data type of `INTEGER`, the search form will contain operators including `Equals`, `Does not equal`,

for that specific field. By using both `datacapture.cfg` and `mdc_dd.cfg`, the search CGI ensures that each field in a search form will always contain the appropriate set of operators from which an end user can choose.

See “Configuring Metadata Search” on page 226 for more information.

3. The metadata search CGI displays the search form on the client system using the GUI.
4. An end user fills in search parameters and submits the search form back to the metadata CGI.
5. The metadata search CGI constructs the appropriate query statements and queries the database.
6. The search results are displayed in the TeamSite GUI.

## Configuring Metadata Search

You must perform two main activities to configure metadata search:

1. Ensure that metadata capture and DAS are synchronized.
2. Add a **Search Metadata** item to the TeamSite GUI so that end users can access metadata search.

The following sections describe these steps in detail. Additional configuration information is included in subsequent sections in case you need to customize the metadata search form or other characteristics of metadata search.

### Synchronizing Metadata Capture and DAS

Metadata search relies on correct synchronization of metadata capture and DAS. If these two features are not synchronized, metadata search will not run correctly. The issue is as follows:

When you configure DAS, you execute the `iwsyncdb.ipl -initial` command. This command generates several files, including `mdc_dd.cfg`. The `mdc_dd.cfg` file is in turn based on information from `datacapture.cfg`. The `datacapture.cfg` file must be configured specifically for metadata capture at your site to ensure that `mdc_dd.cfg` is generated correctly for use with metadata capture and search at your site. Therefore, it is *essential* that `mdc_dd.cfg`

be generated *after* you set up `datacapture.cfg` in `iw-home/local/config` as described earlier in this chapter. There are two ways to ensure that this is the case:

1. Configure metadata capture as described earlier in this chapter before configuring DAS as described in the *DataDeploy Administration Guide*, or
2. Execute the following command if DAS was already configured prior to your configuring `iw-home/local/config/datacapture.cfg`:

```
iwsyncdb -mdcddgen [-force]
```

You can execute this command whenever you need to resynchronize DAS and metadata capture/search. See the *DataDeploy Administration Guide* for details about `iwsyncdb` usage.

Regenerating `mdc_dd.cfg` overwrites the existing version of the file, including any changes you might have made to it.

## **Adding Metadata Search to the TeamSite GUI**

Because metadata search is an area-specific feature, it is recommended that end users access it via the **View** menu in the TeamSite GUI. To add a **Search Metadata** item to the TeamSite GUI's **View** menu, add the following line to the [`iwcgi`] section of `/etc/iw.cfg`:

```
custom_menu_item_searcha="View", "Search Metadata", "iwsearchmetadata.cgi",
"all", "scrollbars=yes,resizable=yes,width=640,height=545"
```

Due to space limitations, this line appears to wrap. The line in the configuration file must not wrap.

The preceding line specifies the following:

- The TeamSite GUI menu (**View**) to which the item will be added.
- The name of the new item (**Search Metadata**).
- The CGI (`iwsearchmetadata.cgi`) that will execute when the item is selected.
- Which users (`all`) can see the menu item.
- The appearance and behavior of the window in which the CGI runs.

See “Custom Menu Items” on page 134 for more information about adding and enabling custom menu items.

## Changing Valid Search Paths

Valid paths for metadata search are set in the [valid\_search\_paths] section of *iw-home/etc/iw.cfg*. By default, all paths are searchable. You can use regular expressions to specify that only certain paths are searchable. See comments in *iw.cfg.example* for more information.

## Making Individual Fields Non-Searchable

You can specify whether any field in a DCT is searchable. By default, all fields are searchable. To make a field non-searchable, specify `searchable="f"` the `<database>` element for that field in *datacapture.cfg*. If you make this change after *mdc\_dd.cfg* was generated, you must regenerate it via the *iwsyncdb -mdcddgen* command.

It is not advisable to edit *mdc\_dd.cfg* itself. Such action could result in inconsistencies between DAS and metadata capture and search.

## Chapter 7

# Managing the TeamSite Server

---

The following topics are described in this chapter:

- Checking Server Status (page 230)
- Reviewing TeamSite Logs (page 233)
- Monitoring the Server Load (page 234)
- Starting and Stopping the Server (page 234)
- Managing the OpenAPI Server (page 234)
- Reconfiguring iwwebd to Recognize a New IP Address (page 235)
- Re-Encrypting User Authentication Information (page 235)
- Troubleshooting (page 236)
- Managing Server Resources (page 242)

# Checking Server Status

## Verifying Server Operation

Verify that the TeamSite server is running correctly by typing:

```
% /bin/ps -ef | grep iwserver | grep -v grep
```

You will see a response similar to this:

```
root 18309 18304 1 08:03:10 ? 2:56 /usr/iw-home/bin/iwserver
local/iw-store
```

If you do not see this response, the main TeamSite process is down. Check the `iwserver.log` log file to see what happened. If TeamSite seems to have died abnormally, run

```
% /etc/init.d/iw.server stop
```

to try to unload the kernel module. On Solaris, you can also run:

```
% modinfo | grep wfs
```

If anything returns, wfs is loaded. The number in the first column is the module id.

If wfs is loaded, stop the server with:

```
% /etc/init.d/iw.server stop
```

Finally, attempt to start TeamSite with:

```
% /etc/init.d/iw.server start
```

## Checking for Multiple Servers

Only one server process should be running at any given time. If there are no processes running, then the server is not running. If more than one server process is running, there is a problem with the server and it should be restarted.

To reset the server to ensure that only one server process is running:

1. Issue the following command to stop the server:

```
% /etc/init.d/iw.server stop
```

2. Verify that all server processes have stopped. If not, manually kill any remaining processes. For more information on the `kill` command, consult a UNIX reference manual.

3. Restart the server with the following command:

```
% /etc/init.d/iw.server start
```

## Checking Request Handling

Verify the server is answering requests correctly by issuing the command:

```
% iw-home/bin/iwversion
```

You will see a response similar to this:

```
iwserver: 5.5.1 Build 6038 Interwoven 20010420
```

If the server does not respond or stops, then the server is not handling requests correctly. Restart the server, as described above.

## Verifying the Server Mount

Verify the server is mounted to the correct drive partition with the command:

```
% df -k | grep iwserver
```

The output should contain a line similar to this:

Filesystem	kbytes	used	avail	capacity	Mounted on
servername:/iwserver	3141968	1542472	1285336	55%	/iwmnt

If the server does not respond properly, restart it as described above.

## Finding the Installation Directory

To find the TeamSite installation directory, use the `iwgethomeCLT`.

### *Usage:*

```
iwgethome [-h | -v | -o]
```

- |    |                                         |
|----|-----------------------------------------|
| -h | Displays usage message.                 |
| -v | Displays version.                       |
| -o | Returns original factory setting value. |

### *Example:*

```
% iwgethome
```

returns

```
/usr/iw-home
```

## Reviewing TeamSite Logs

TeamSite records events in TeamSite log files as described below.

<b>Log file</b>	<b>Default location</b>	<b>Contents</b>
Installation log	<code>iw-home/install/iwinstall.log</code>	Record of the TeamSite installation process.
Server log	<code>/var/adm/iwserver.log</code>	Record of the state of TeamSite over time. Tracks when the TeamSite server is started, stopped, mounted, etc. The location of this file is contained in <code>/etc/defaultiwlog</code> . You can also find this file using the CLT <code>iwgetlog</code> .
Trace log	<code>/var/adm/iwtrace.log</code>	Record of any irregularities on the TeamSite server. Used by Interwoven Professional Services to diagnose system performance issues. The location of this file is contained in <code>/etc/defaultiwtrace</code> . You can also find this file using the CLT <code>iwgettrace</code> .
Event log	<code>/var/adm/iwevents.log</code>	Record of activities on TeamSite. Tracks when files are submitted, published, branches created, etc., including DiskLow, Freeze, ShutDown, StartUp and Thaw events. Used with TeamSite triggering scripts. The location of this file is contained in <code>/etc/defaultiwevlog</code> . You can also find this file using the CLT <code>iwgetelog</code> .
Workflow log	<code>/var/adm/iwjoberrors.log</code>	Record of output from workflow runtime diagnostics.

## Monitoring the Server Load

The TeamSite CLT `iwstat` returns a list of all current TeamSite processes. See *TeamSite Command-Line Tools* for details about `iwstat` usage and output.

## Starting and Stopping the Server

To stop the TeamSite server:

```
% /etc/init.d/iw.server stop
```

To restart the TeamSite server:

```
% /etc/init.d/iw.server start
```

## Managing the OpenAPI Server

The OpenAPI server is part of the Interwoven Servlet Engine and is automatically installed and started by the TeamSite installation program. This differs from earlier releases of TeamSite where OpenAPI existed as a separate service. The current architecture reduces inter-process communication on the TeamSite server, resulting in improved performance.

Additionally, an OpenAPI configuration file, `openapi.cfg`, is also installed by the TeamSite installation program. It does not need to be modified. I18N requirements are also automatically handled by the installation program, which creates the directory `iwopenapi/locale`.

### Verifying that the OpenAPI Server is Running

To verify that the OpenAPI server is running, issue the following command:

```
% ps -ef | grep iwservletd
```

The system will return the Interwoven Servlet Engine daemon that is running. If it is not running, restart it as described in the next section.

## Starting and Stopping OpenAPI

To manually restart the Interwoven Servlet Engine issue the following CLT:

```
% iwreset -a
```

This command causes `iwserver` to re-read its configuration from the `iw.cfg` file, and restart (stopping if necessary) `iwwebd`, `ipproxy`, and `iwsvrletd`.

**Note:** Do *not* use the old OpenAPI startup script `/etc/init.d/iw.openapi`.

If you would like more information about OpenAPI, consult the administration chapter of the *OpenAPI Developer's Guide*, available online as part of the OpenAPI SDK.

## Reconfiguring iwwebd to Recognize a New IP Address

If you change the IP address of the server, you need to reconfigure `iwwebd` to recognize the new address. To do this:

1. Go to the `iwwebd.bin` directory.
2. Run `iwwebd_conf.ip1`.
3. Restart `iwwebd`.

`iwwebd` will be reconfigured and the `iw.cfg` file will be updated with the new IP address.

## Re-Encrypting User Authentication Information

The TeamSite CLT `iwsessionkeygen` generates the key used to encrypt user authentication information. Running this command invalidates all current user sessions and generates a new key. You may want to run `iwsessionkeygen` periodically to protect system security.

To generate a new encryption key for user authentication information, log in as root and run `iwsessionkeygen`. All user sessions will need to log in again to continue working in any TeamSite interface.

Since WebDesk Pro does not allow users to easily resume interrupted sessions, it is recommended that you do not run `iwsessionkeygen` while users are working in WebDesk Pro. If you do run `iwsessionkeygen` while users are working in WebDesk Pro, these users may experience a variety of errors (for example, failure to connect to the TeamSite server, or invalid session strings). If this happens, WebDesk Pro users should log out, then log in again.

## Troubleshooting

Interwoven's Professional Services and Technical Support can assist you with any installation and configuration issues you might encounter. You can also consult the Interwoven Support Knowledge Base, available at <http://support.interwoven.com>.

### Repairing the Backing Store

The following section contains information about the backing store repair tools provided with TeamSite. If you are experiencing problems with the TeamSite backing store such as missing TeamSite areas or missing file versions, use `iwfsck` to check the backing store. You can use `iwfsck -y` and `iwfsfix` to repair the backing store depending on the results of your backing store check.

**iwfsck**

Diagnoses backing store problems and allows repair of some of the problems found.

**Usage:**

```
iwfsck [-h] [-v] [-x|-xx|-xxx] [-l] [-y] [-b path] [-z]
[-d [[-s] | [-f] [-m] [-p]] [-r]]
[-o file] [-e file] [-t file] [-u file] [vpath]
```

-h	Displays usage message.
-v	Displays version.
-x	Requests extra output and increments verbosity level. Prints additional information about what <i>iwbsck</i> is doing as it operates. Each x increments the verbosity level by 1. The highest level of verbosity is level 3 (-xxx). In the higher levels of verbosity, an extremely large quantity of output may be produced.
-l	Prints output as HTML. This option is used by the <i>iwfsckcgi.cgi</i> program.
-y	Repairs damaged files while running. In this mode, damaged files are deleted while <i>iwfsck</i> is running. The TeamSite server must be down when specifying this option. If the TeamSite server is running when this option is specified, a warning displays and this option is ignored.
-b <i>path</i>	Uses <i>path</i> as the backing store location. The default is the configured backing store location returned by <i>iwgetstore</i> for the TeamSite server.
-z	Checks events in branches.
-d	Checks directories and files in addition to the normal checking of branches and areas. All directories and files from the <i>vpath</i> are walked. If a <i>vpath</i> is not specified on the command line, the walk begins at the / <i>vpath</i> .

The following options are only allowed when -d is specified:

- f Provides a fast reference check (not allowed with -p, -m, or -s). All references from the root are walked aggressively looking for missing references. If a missing reference is found, that part of the tree is marked *suspect*, and a more expensive walk with vpaths is done on that part of the tree to determine the directories and files affected by the problem.
- s Provides a stack walk, which is slower but uses less memory than the default (not allowed with -f). This mode uses the least amount of memory, but it is the least efficient for walking the entire tree of files and directories from the root.
- m Checks ModLists for directories. A ModList is a data structure that is a shadow tree to the directory structure within a workarea. This shadow tree allows the modified files within a workarea to be determined quickly without having to traverse every file and directory within a workarea.
- p Checks protopaths. A protopaths is a data structure that allows file names and history information to be determined without the expense of walking up to the root of an area through directories; however, it can be expensive.
- r Checks parents. Parents and anti-parents are the reference counting mechanism used by the TeamSite server. If zero parents are found for a file, it indicates a problem. It can be expensive.

The following options specify where output goes (note that `stdout` and `stderr` may be redirected in the normal way in a command line shell and that `-o` and `-e` are provided to allow redirection when shell redirection is not available):

<code>-o file</code>	Specifies output file for server startup information.
<code>-e file</code>	Specifies the file to write error messages to.
<code>-t file</code>	Specifies the file to write reports to.
<code>-u file</code>	Specifies the summary file.
<code>vpath</code>	Specifies the starting vpath to walk directories when <code>-d</code> is used.

### **Examples:**

To check areas and branches, issue the command:

```
% iwfsck
```

To check directories and files in addition to branches and areas, issue the command:

```
% iwfsck -d
```

Use the following command to check protopaths and parents in addition to branches, areas, directories, and files. This command can be very resource intensive.

```
% iwfsck -d -p -r
```

## **iwfsckcgi.cgi**

This program provides an optional GUI interface to run **iwfsck**. You can access this interface through a browser:

***server\_name/iw-bin/iwfslogin.cgi***

You will be prompted for the root or Administrator password. Once you have been authenticated, a screen will provide two choices:

- Perform content recovery
- Perform backing store check

To run **iwfsckcgi.cgi**, select **Perform backing store check**.

## **iwfsfix**

If **iwfsck** finds problems that cannot be repaired or the TeamSite server is running when the backing store is diagnosed, it outputs lines in the format:

**FIX iwfsfix repair args**

The repairs and their arguments are shown below. To perform necessary repairs, copy the **FIX** line issued by **iwfsck** and paste it on the command line, with the word **FIX** removed.

There are also repairs for ModLists that must be performed when the TeamSite server is running. On UNIX, these lines are in the format:

**FIX /bin/touch junkfile; /bin/rm junkfile**

*junkfile* is a uniquely named file that is created and removed from an affected directory.

The repairs that can be performed with **iwfsfix** are:

**delete\_tag branch\_id tag\_id**

Removes the reference to a tag (lock) from a branch. This is done when the tag point itself is missing.

`delete_tag_and_point branch_id tag_id`

Deletes the reference to a tag (lock) from a branch and removes the tag point itself. This is usually done when a tag duplicates or conflicts with another tag within a branch.

`delete_direntry directory_id diritems_index filename`

Deletes the directory entry for a damaged or missing file.

`replace_direntry directory_id diritems_index filename new_standin_id`

Repairs a directory entry to point to a correct standin ID.

`delete_area area_id`

Deletes the point for an area.

`delete_area_from_branch branch_id area_id workarea | edition`

Deletes the reference to an area (workarea or edition) from a branch. This cannot be done on a staging area because a branch by definition always contains a staging area.

`null_previous point_id`

Sets to null (-1) the PreviousPoint reference within a point. This is done when the PreviousPoint reference for a point is incorrect.

`clone_diritems directory_id diritems_index new_gen_id new_dot_dot`

Clones a set of directory items within a directory to create a new set. This is done when a set of directory items is shared between areas, but it should not be shared.

# Managing Server Resources

## Disk Space

### TeamSite Data Store

The location of the TeamSite backing store is contained in `/etc/defaultiwstore` (alternatively, you can use the CLT `iwgetstore` to find this location). This file can only contain one address at a time. Changing the address contained in this file changes the location of the backing store.

To move the TeamSite backing store:

1. Shut down the server.
2. Move the entire backing store to the new location.
3. Edit `/etc/defaultiwstore` so that it now points to the new location. **Note:** Never modify `/etc/defaultiwstore` while the TeamSite server is running.
4. Restart the TeamSite server.

To create a new, empty, TeamSite backing store:

1. Shut down the TeamSite server.
2. Edit `/etc/defaultiwstore` to point to an empty directory on a partition with enough space for the TeamSite backing store.
3. Restart the TeamSite server.

TeamSite will create a new (empty) backing store when you restart the server. The new backing store will contain only a main branch, which will have an empty initial edition, a staging area, and no workareas. You can populate this new backing store with content just as you did your old one. If you point `/etc/defaultiwstore` back to the older backing store, all the content contained in the older backing store will reappear.

## Checking Disk Space Usage

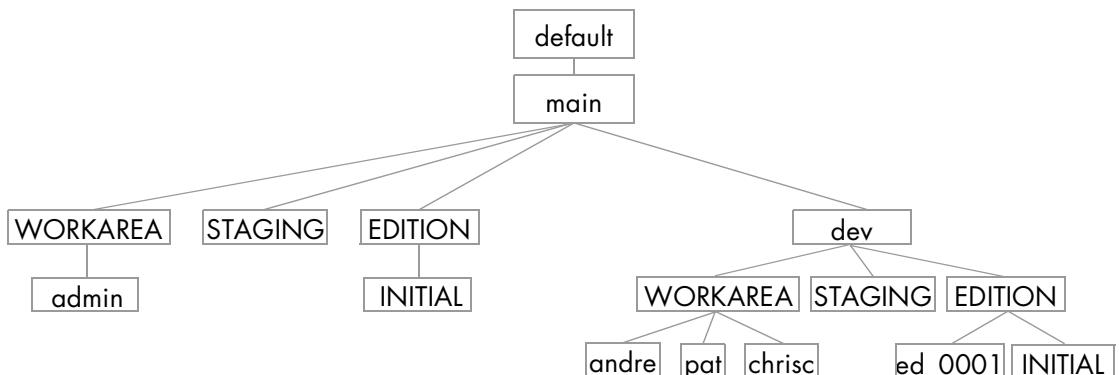
When checking disk space usage, you can use

```
% df -k `cat /etc/defaultiwstore`
```

to check the size of the directory contained in `/etc/defaultiwstore` or `/iwmnt`. That is, you can check the size of either the backing store *or* the mount point. Although the mount point contains many virtual copies of files in workareas, staging areas, and editions, `df -k` will only check the actual disk space used.

## TeamSite File System Mount

The TeamSite file system mount contains a file system view of all the branches, workareas, staging areas, and editions on the TeamSite server. TeamSite areas do not contain physical copies of the entire Web site, but rather pointers to the files contained in the Web site. The only physical files contained within TeamSite areas are the files that have actually been modified in those areas. That is, the only files actually contained in a workarea are those files that have been modified in that workarea but not yet submitted; the only files contained in the staging area are the files that have been submitted since it was last published; the only files in an edition are the files that have changed since the previous edition was published.



*Sample TeamSite file system structure*

Each branch contains three directories: WORKAREA, containing all the workareas on the branch; STAGING, containing the staging area for the branch, and EDITION, containing all editions on

the branch. It may also contain directories that hold sub-branches. In the example above, the main branch (`main`) contains one workarea, a staging area, an initial edition, and a sub-branch (`dev`). The sub-branch contains three workareas (`andre`, `pat`, and `chrisc`), a staging area, and two editions.

Although many of the files contained within this file system structure are virtual, they can be treated as if they were real. They will appear to exist even when you run links checkers and scripts against them. However, staging areas, editions, and container directories (for example, `WORKAREA`, `EDITION`, `main`, or `dev`) are all read-only. Only workareas can be written to.

## **Recovering Disk Space**

To reclaim some disk space, you can delete old editions, which will delete all files actually contained in that edition, in addition to all intermediate submissions between publication of editions.

## **Routine Maintenance: Metadata Forking**

Metadata forking conserves disk space by reducing the number of files whose content is duplicated throughout the TeamSite backing store. That is, if you have an old version of a file in one branch, and an identical file version on another branch, the same data may appear twice in the backing store. Metadata forking eliminates this type of duplication. This operation results in no user-visible changes to the TeamSite virtual file system. For example, file histories are unchanged.

To use metadata forking, run the `iwfsshrink` utility. The `iwfsshrink` utility may be run while the TeamSite server is running; however, TeamSite may experience some performance degradation while it is running. Also, `iwfsshrink` may not remove all duplicates (for example, it will not remove any duplicates created by TeamSite users while the utility is running).

1. Issue the `iwfsshrink` command:

```
% iwfsshrink run
```

2. The utility may take several hours to run. Use the `status` option to view the current status. You can also pause the operation with the `pause` option, then restart it with the `run` option. See “`iwfsshrink Syntax`” for a full list of options.

The `iwfsshrink` utility should be run every few months.

### ***iwfsshrink Syntax***

```
iwfsshrink [-h] [-v] [run | pause | abort | status]
```

-h	Displays usage message.
-v	Displays version.
run	Starts the <code>iwfsshrink</code> process.
pause	Temporarily stops the <code>iwfsshrink</code> process. It can be restarted with the <code>run</code> option. Because <code>iwfsshrink</code> takes a long time to run, you may want to start it during off-hours. When activity increases, you can pause it until the next period of inactivity.
abort	Terminates the <code>iwfsshrink</code> process.
status	Shows information about the latest <code>iwfsshrink</code> process.

**Examples:**

```
% iwfsshrink status
```

when `iwfsshrink` has finished running, returns a message similar to:

```
Not currently running.
Last started Mon Jun 26 15:47:53 2000
Last completed Tue Jun 27 00:40:04 2000
Files examined: 317974
Bytes examined: 75936814830
Files found to be duplicates: 233430
Files converted: 198352
Bytes removed: 23455046531
```

**Moving the Backing Store and Removing Old Versions**

If you are running out of disk space and `iwfsshrink` doesn't recover enough extra space, you might need to move the TeamSite backing store (see page 242). The TeamSite backing store must reside on a single logical volume, e.g., a single disk or an array of disks.

Alternatively, if you have unused branches in TeamSite, you can delete these branches to recover disk space.

Over time, individual branches take up an increasing amount disk space, as the number of versions and files on the branch grows. If you do not need any of your old version history, you can create a new (empty) branch, create a workarea, copy all the old content into the workarea, then delete the old branch. Exercise extreme caution when doing this, as all versioning and metadata information will be irrevocably lost.

## Chapter 8

# TeamSite Backing Stores

---

This chapter describes the TeamSite backing store functionality including creating multiple backing stores (either by converting an existing backing store or creating new backing stores). It also includes what you need to know to prepare for conversion. The information is organized as follows:

- Backing Store Overview
- Planning the Backing Store Conversion
- Converting Backing Stores Using the GUI
- Converting Backing Stores from the Command Line
- Creating Multiple Backing Stores
- Administration CLTs
- UID Changes to the TeamSite Backing Store

## Backing Store Overview

The backing store is a large directory structure created by the TeamSite installation program that contains TeamSite files and metadata. By default, the backing store is located in `/local/iw-store`.

Previous releases of TeamSite have been limited to one backing store per TeamSite server. This release supports as many as eight backing stores per TeamSite server. These backing stores can be located on different file systems, local to the TeamSite server machine. The functionality that enables multiple backing stores is known as *MultiStore*.

To include MultiStore support in TeamSite (and improve overall performance), a new backing store format needed to be implemented. This format is used by all backing stores created using the current TeamSite release. If you have a backing store created with TeamSite version 4.5.x or 5.0.x, you must convert the old backing store to use the new format as described in either “Converting Backing Stores Using the GUI” on page 252 or “Converting Backing Stores from the

Command Line” on page 256. You can also use this procedure to divide your single old-format backing store into multiple new-format backing stores.

Dividing your existing backing store into new multiple stores (possibly on different file systems) enables you to simplify data management, including faster data backup. It also avoids having your backing stores grow to unmanageable sizes.

You can migrate data to your new stores any way you choose, but the data between the stores is completely independent and may not be migrated to other stores using inter-branch copying. Copies remain branch-specific and cannot be used at the backing store level.

Backing stores have a corresponding archive in the VPATH. In previous versions of TeamSite, there was only one archive named `default` with a corresponding backing store called `iw-store/default`. MultiStore functionality allows for multiple backing stores with user-assigned names. Each backing store is similar to the `default` archive in that it contains a single root branch called `main` and is independent of any other store controlled by the server. All mounted backing stores are assigned a unique store ID number and maintain their own set of inodes that are stored persistently inside each backing store.

Backing stores which are named using multibyte characters must be created by editing the `iw.cfg` file. For detailed information, see “Defining Backing Stores in the `iw.cfg` File” on page 262.

## Planning the Backing Store Conversion

Before you begin the backing store conversion, read through the conversion overview, and ensure you have satisfied the conversion prerequisites before beginning the actual step-by-step conversion procedure as described in the following sections:

- “Converting Backing Stores Using the GUI” on page 252
- “Converting Backing Stores from the Command Line” on page 256

Also note that there is a *Backing Store Conversion Guide* available on <http://support.interwoven.com> that contains the latest conversion information.

## Conversion Overview

This section describes the conversion procedure in very general terms. It is intended to help you understand what is involved in the conversion procedure before you begin.

1. Satisfy the prerequisites (as described on page 251).
2. Decide how you want to organize your new backing store on the target system:
  - Convert your single old-format backing store into a single new-format backing store.
  - Convert your single old-format backing store into multiple new-format backing stores. You can later convert your single new-format backing store into multiple new-format backing stores by using the `iwmigrate` CLT as described on page 270).

Your conversion options are depicted in the graphic at the end of this section.

3. Run `iwfsck -d` on your source backing store to prepare for conversion. The `iwfsck` CLT is described in the *TeamSite Command-Line Tools* manual.

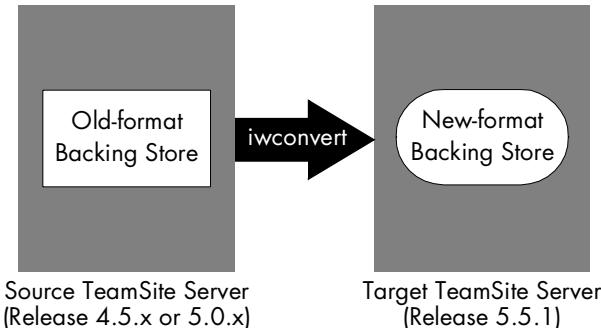
4. Run the `iwconvert` conversion tool from either the GUI or the command line.

You may repeat this step any number of times depending on what you plan to convert from your existing store, and whether you plan to create multiple new-format stores.

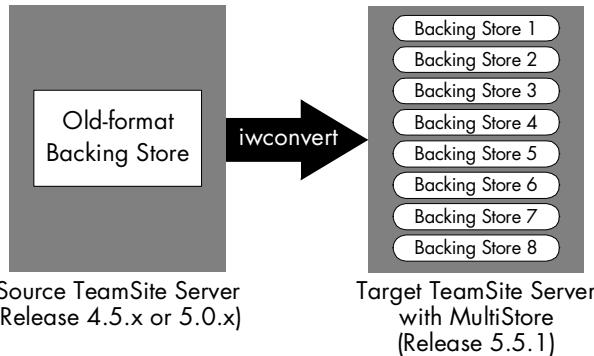
### Optional Steps:

5. Make an edition of your staging area after the initial conversion pass (or passes) is complete. This edition will contain anything submitted while the initial conversion was running. It should be much smaller than your other editions and should convert faster.
6. Run the `iwfreeze` CLT to prohibit any more changes to the staging area.
7. Create and convert the final edition.
8. If you want to use the source system (where your current TeamSite 4.5.x or 5.0.x installation resides) as your production server *after* the conversion is complete, you will need to install TeamSite 5.5.1 on this server and copy the converted (new-format) backing stores onto this server.

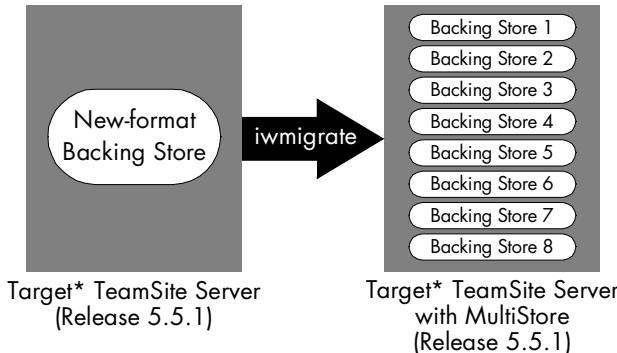
## Old-format Single Store → New-format Single Store



## Old-format Single Store → New-format Multiple Stores



## New-format Single Store → New-format Multiple Stores



\* The iwmigrate procedure can be performed on one TeamSite Server; iwconvert requires two systems.

## Conversion Prerequisites and Tips

- You must have two systems running the same operating system (they do not need to be the same version) and be located on the same network. This document refers to these as the *source* system (which contains your current, old-format backing store) and the *target* system (which will host the new-format version of the backing store).
- Ensure that TeamSite 4.5.x, or 5.0.x is installed and licensed on the source system.
- Back up your existing installation and backing store.
- Ensure that TeamSite 5.5.2L is installed and licensed on the target system.
- The target system must have at least as much disk space as the source system.
- Consider creating separate backing stores for branches that meet the following criteria:
  - Distinct deployment targets
  - Legacy or infrequently accessed
  - Distinct ownership within your organization
- Decide how you are going to divide the source backing store for conversion. Depending on the size of your source backing store, and the organization of your TeamSite implementation, you may choose to convert a range of editions, a single edition, or branch-by-branch.
- Before running the `iwconvert` command, run `iwfsck -d` on your source backing store to prepare for conversion. The `iwfsck` CLT is described in the *TeamSite Command-Line Tools* manual.
- Allow plenty of time for the conversion to complete. While there are many variables affecting the time it takes to convert a backing store, tests show that the conversion runs at approximately 500 megabytes (MB) per hour. Note that this is a very general number and you should not be concerned if your conversion runs at a different rate.
- Plan to publish new editions of the staging area if you allow users to submit files while the conversion is running. You may have to do this multiple times. Eventually, you will need to freeze your source server to prohibit users from using TeamSite during the final conversion.
- You can use the `iwmigrate` CLT to migrate data between new backing stores. You can also use your operating system's copy functionality, but you will lose history and version information.
- You can have a maximum of eight active backing stores on your target system. You can have more than eight backing stores, but only eight can be active.
- You can create backing stores on NFS-mounted remote servers if you are using disk management devices.

- Workflow tasks must be completed prior to conversion, or you must run the `iwwfconvert` CLT to copy the active, unfinished workflow to your new backing store as the final conversion step.
- To avoid having the backing store conversion procedure create an unknown user on the target system, the user doing the conversion must have root access on both systems. If an unknown user is created, you can use the `iwidmap` CLT to remap the unknown user to an appropriate user on the target system.
- Your backing store represents a major investment to your organization. If for any reason you feel that you want help with the conversion process, please contact Interwoven Client Services (<http://www.interwoven.com/services>).

## Converting Backing Stores Using the GUI

Whether you use the conversion GUI or convert from the command line, the backing store conversion is done by the `iwconvert` program. This utility can be run directly from the command line (as described in “Converting Backing Stores from the Command Line” on page 256), from a script generated by the conversion GUI, or interactively by using the conversion GUI. The bulk of the conversion is done while your existing TeamSite server is running, but you will need to freeze your source server (or prohibit users from using TeamSite) for the final few editions to ensure that no data is lost.

The backing store conversion GUI is a CGI program (`iwconvert.cgi`) that is installed by the TeamSite installation program and displayed in your Web browser. It is supported by a process called `iwconvertserver` that communicates with remote TeamSite servers and invokes `iwconvert` on behalf of the GUI. You must run the `iwconvertserver` process manually for the GUI to function properly.

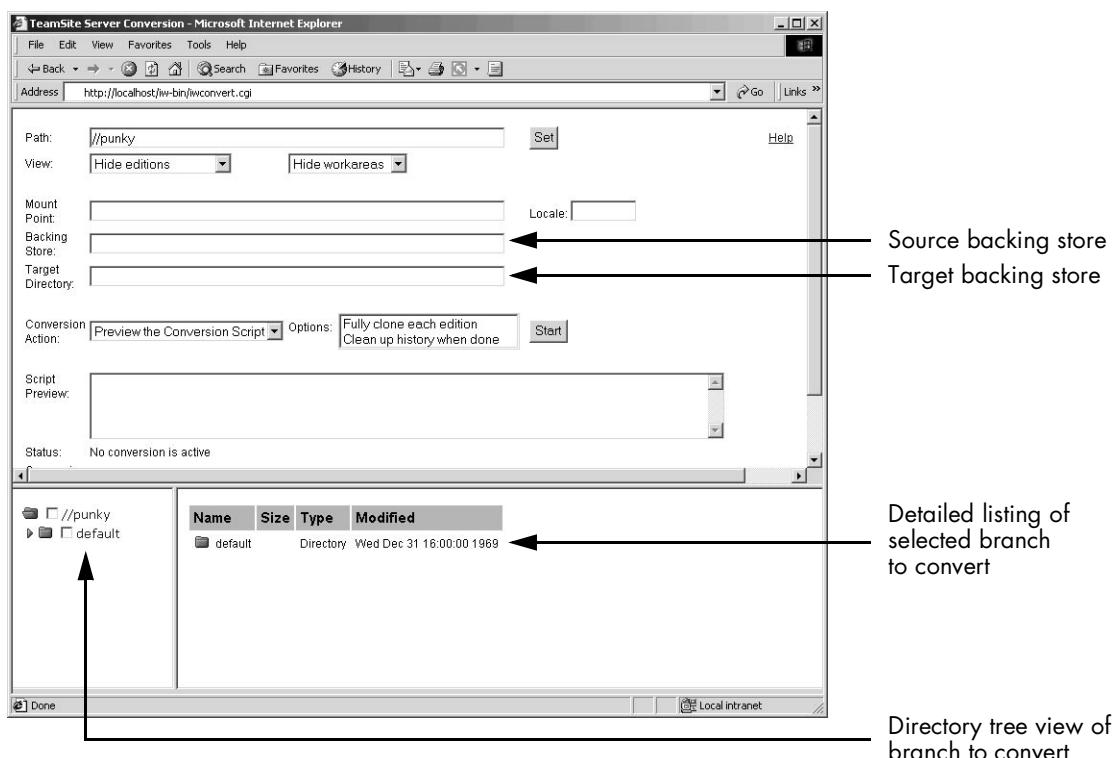
Complete the following procedure to convert your old-format backing store to use the new backing store format:

1. Log into the TeamSite 5.5.1 system as root.

2. Run the **iwconvertserver** utility directly from the command line in **iw-home/bin** with no arguments.
3. Open your Web browser.
4. Type **http://localhost/iw-bin/iwconvert.cgi** in the **Address** field (Internet Explorer) or the **Location** field (Netscape).

**Note:** If you are not working on the same computer that will contain the new backing store, you must specify the system name instead of `localhost` in this step. For example, if you wanted the new backing store to be located on a server named `factotum`, type:  
**http://factotum/iw-bin/iwconvert.cgi**

The conversion GUI is displayed in your default browser:



5. In the **Path** field, enter the network version path (**vpath**) to the TeamSite server containing the source backing store.

If the source backing store is located on a remote server you must begin the path with two forward slashes (//). For example, in the previous graphic, the old-format backing store to be converted is located on a remote server named punky.

6. Click **Set** to display the source tree of the **vpath** of the source backing store.
7. Set the **View** menus to filter the files that are displayed in the directory tree view of the source backing store (lower left frame in the conversion GUI). The options are:
  - **Hide editions**—Individual editions are not displayed. Converting this branch converts all corresponding editions.
  - **Show editions**—Displays individual editions. You must select which editions are converted. (Expanding an EDITIONS directory containing a large number of editions can be very slow.)
  - **Show recent editions**—Displays the 10 most recent editions. You must select which editions are converted.
  - **Hide workareas**—Workareas associated with the selected branch are not displayed or converted.
  - **Show workareas**—Individual workareas are shown. You must select which workareas are converted. (Expanding a WORKAREAS directory containing a large number of workareas can be slow.)
8. Click **Set** to update the source tree if you modified the **View** menus (in step 7).
9. In the **Mount Point** field, enter the file system path to the TeamSite virtual file system of the source machine, for example, /iwmnt.
10. In the **Locale** field, enter the locale of the backing store to be converted, or leave the field blank to use the default locale.

This entry is used to specify how non-ASCII metadata is interpreted. The following are valid locales:

- iso-8859-1      • iso-8859-2      • iso-8859-3      • iso-8859-4      • iso-8859-5
- iso-8859-6      • iso-8859-7      • iso-8859-8      • iso-8859-9      • iso-8859-15
- euc-jp            • euc-tw            • euc-cn            • euc-kr            • shift\_jis
- big5              • gb2312            • utf-8            • utf8

11. In the **Backing Store** field, enter the file system path to the source backing store, for example: /net/punky/iw-store/default.

This field is optional, but providing the path results in a significantly faster conversion.

12. In the **Target Directory** field, enter the name of the directory where the new backing store is to be created, for example: /local/iw-store/newStore1.

13. Select one of the following actions from the **Conversion Action** menu:

- **Preview the Conversion Script**—Displays the iwconvert command that is generated based on the selected branches and options in the Script Preview field when **Start** is clicked.
- **Generate the Conversion Script**—Displays the iwconvert command that is generated based on the selected branches and options in the Script Preview field, and writes the command to a script file when **Start** is clicked.

The script files use the iwconvert\_#.sh naming convention (where # represents an integer) and is created in the iw-home/local/logs directory on the target system.

- **Run Conversion**—Displays the iwconvert command that is generated based on the selected branches and options in the Script Preview field, writes the command to a script file, and invokes the iwconvert command when **Start** is clicked.

14. Select none, one, or both of the following iwconvert options (use Shift+click to select both options):

- **Fully clone each edition**—Runs iwconvert with the -f option, so that editions are cloned without any submit event history. Use this option if there are gaps in the set of editions that you are converting, or if you do not want to save the submit history.
- **Clean up history when done**—Runs iwconvert with the -c option to generate submit events correctly.

If you do not select one or both of the options, all history is copied.

15. Click **Start** to initiate the action defined in step 13 and step 14 and to display the status of the conversion:
  - **Status**—Displays whether or not `iwconvert` is currently running from the GUI. If “`iwconvertserver not enabled`” is displayed, you must run the `iwconvertserver` process that supports the GUI (described in “Administration CLTs” on page 267).
  - **Conversion Step**—Displays the current `iwconvert` command line if it has been started from the GUI.
  - **Conversion Detail**—Displays the progress of the conversion if `iwconvert` has been started from the GUI.
16. If files are submitted during the conversion procedure, you will need to freeze the server (by using `iwfreeze`), create a new edition that contains these files, and repeat the conversion procedure.

**Note:** The `iwconvert` program creates temporary workareas (`temp_workarea`) in each converted branch as a by-product of the conversion process. You should manually delete these after the conversion.

## Converting Backing Stores from the Command Line

The TeamSite installation program installs a set of command-line tools in the `iw-home/bin` directory. All of these tools are documented in the *TeamSite Command Line Tools* manual that corresponds with your platform. For convenience, the new CLTs are also included in this document. The `iwconvert` CLT is described in this section the other new CLTs are described in “Administration CLTs” on page 267.

### **iwconvert Command-Line Tool**

The `iwconvert` CLT converts old-format (TeamSite 4.5.x and 5.0.x) backing stores to the new high-performance backing store format.

Complete the following steps to optimize the conversion process.

- Upgrade your source machine to TeamSite 4.5.1 Service Pack 2, or TeamSite 5.0.1 Service Pack 2 (or higher), with all available patches.

- Ensure you have the most recent version of the `iwconvert` tool.  
Updates to the `iwconvert` and `iwmigrate` tools shipped with TeamSite 5.5.1 will be available on the Interwoven support website. Before using either CLT, check the Interwoven support website (<http://support.interwoven.com>) to ensure you have the most recent version of each tool.
- Before running the `iwconvert` command, run `iwfsck -d` on your source backing store to prepare for conversion. The `iwfsck` CLT is described in the *TeamSite Command-Line Tools* manual.

## Options

The following options are valid for the `iwconvert` command:

<code>-h</code>	Displays the usage message.
<code>-v</code>	Displays the version number.
<code>-b branch_vpath</code>	Location of the branch that contains the editions or workareas to be converted. If the <code>vpath</code> begins with <code>//hostname/</code> the branch is located on a remote TeamSite server. <b>Note:</b> Branches are converted recursively—all editions in subbranches under the specified branch are also converted unless the <code>-d</code> option is specified.
<code>-c</code>	Cleans up the history information of a previously converted backing store. Requires that <code>-o</code> is also specified. This <code>iwconvert</code> step must be performed last, as a separate step, because it may have interbranch dependencies. Note that this action may safely be executed multiple times.
<code>-d</code>	Do <i>not</i> recursively convert subbranches.
<code>-s starting_edition</code>	Specifies the first edition in a range of editions to be converted for the specified branch (the default is INITIAL). Requires that <code>-b</code> , <code>-d</code> , <code>-o</code> , and <code>-m</code> are also specified.

<b>-e</b> <i>ending_edition</i>	Specifies the last edition in a range of editions to be converted for the specified branch (the default is the most recent edition). Requires that <b>-b</b> , <b>-d</b> , <b>-o</b> , and <b>-m</b> are also specified.
<b>-f</b>	Forces a full clone of each edition <i>without</i> the history of submit events for the editions. Use this option if there are gaps in the set of editions that you are converting, or if you do not want to save the submit history.
<b>-l</b> <i>locale</i>	Specifies the native locale of the backing store being converted and how non-ASCII metadata is interpreted. If this option is not specified it defaults to <code>LC_LOCAL</code> .
<b>-m</b> <i>iwmnt_mount_point</i>	Specifies the mount point for the existing (source) <code>iwserver</code> installation. Required with <b>-b</b> , <b>-r</b> , and <b>-w</b> .
<b>-n</b> <i>old_backing_location</i>	Use direct access to the old backing store for faster conversions. Can be used with <b>-b</b> , <b>-r</b> , or <b>-w</b> . If this option is not specified, <code>iwconvert</code> will run slowly due to calls to <code>sci_GetPredecessors()</code> .
<b>-o</b> <i>new_backing_location</i>	Location of the new backing store. This must be a path to the store root. For example, the store named <code>default</code> is specified by: <code>/local/iw-store/default</code> .
<b>-w</b> <i>workarea_name</i>	Converts the specified workarea. Requires that <b>-b</b> , <b>-o</b> , and <b>-m</b> are also specified.
<b>-x</b>	Increases the verbosity level. Maximum verbosity is level 3, expressed as <code>-x -x -x</code> .
<b>Ctrl+c</b>	Stops <code>iwconvert</code> at the end of the edition currently being converted. When you restart the conversion, <code>iwconvert</code> ignores the editions in the branch that have already been converted and converts the remaining editions.

## Usage Summary

- Convert editions:

```
iwconvert -o new_backing_store_location -m iwmnt_mount_point
[-n old_backing_store_location] -b branch_vpath [-d [-s
starting_edition] [-e ending_edition]]
```

- Convert a workarea:

```
iwconvert -o new_backing_store_location -m iwmnt_mount_point
[-n old_backing_store_location] -b branch_vpath -w workarea_name
```

- Clean up history:

```
iwconvert -o new_backing_store_location -c
```

## Example

```
iwconvert -o /local/iw-store/default -m /source/iwmnt -b
//factotum/default/main/intranet
```

## Conversion Procedure

Convert each branch (and its associated subbranches) by completing the following procedure.

- Issue the `mount` command to create a mount point for the source backing store:

```
mount -F nfs source:/local/iw-store /source/iwstore
```

- Issue the `mount` command to create a mount point for the source server:

```
mount -F nfs -o vers=2 source:/iwserver /source/iwmnt
```

- Decide which editions are to be converted.

Branches are converted recursively—all editions in the subbranches under the specified branch are also converted unless the `-d` option is specified. You can convert an individual edition (typically the most recent) or, if the `-d` option is specified, a range of editions.

For example, the range of INITIAL to `ed_0006` would convert seven editions: INITIAL and `ed_0001` through `ed_0006`. Each range of editions converted requires a separate invocation of `iwconvert`.

4. If files have been submitted to the staging area since the last edition was published, publish a new edition.

5. Issue the `iwconvert` command from the `iw-home/bin` directory:

```
iwconvert -o new_backing_store_location -m /mount_location -b source_branch -d -s start_of_edition_range -e end_of_edition_range
```

For example, using the example edition range from step 3, on a remote server named `factotum`, and a branch named `default/main/intranet`:

```
iwconvert -o /local/iw-store/default -m /source/iwmnt -b //factotum/default/main/intranet -d -s INITIAL -e ed_0006
```

**Note:** You should save the `stdout` and `stderr` output from the `iwconvert` procedure to a log file by appending the following to the command in this step:

```
...INITIAL -e ed_0006 > convert.log 2 &>1
```

6. Convert the changes that occurred while the conversion was running by performing either of these steps:

- Submit all changes from workareas to staging, then publish a new edition that contains these changes. Convert this new edition as described in step 5.
- Convert again using the `-w` option to convert the workareas that contain changes not submitted to the staging area before the conversion described in step 5.

```
iwconvert -o /local/iw-store/default -m /source -b //factotum/default/main/intranet -w jerome
```

If you have changes in a large number of workareas it is easier to have users submit their changes and publish and convert a new edition rather than converting the workareas that contain changes.

This step should complete much faster than your original conversion.

7. Freeze your source server by running the `iwfreeze` command from the `iw-home/bin` directory specifying a large number of seconds for the freeze, for example:

```
iwfreeze +50,000
```

8. Repeat step 6 to convert the final changes that were made during the second conversion.

9. Run `iwconvert` with the `-c` option to clean up the second-predecessor links in the new-format (target) backing store, for example:

```
iwconvert -o /local/iw-store/default -c
```

These links are created by TeamSite operations including Copy To. The clean up of history must be done as a separate pass at the end of other `iwconvert` passes because the second-predecessor links can point in various directions between branches and areas in the backing store, and the referenced objects may not be converted at the time they are needed.

If the workareas contain versions of files that have not been converted, the correct contents of those files are copied into the workarea, and those files will show up as modified.

**Note:** The `iwconvert` program creates temporary workareas (`temp_workarea`) in each converted branch as a by-product of the conversion process. You should manually delete these after the conversion.

## Creating Multiple Backing Stores

Multiple backing stores can be created using two different methods depending on where you want to locate them, and whether you want to use multibyte characters in their names.

- `iwstoreadm` CLT—Creates and activates new backing stores when issued with the `-a` option.
  - Accepts ASCII characters for store names.
  - Creates the new backing store in the default location (typically `/local/iw-store/`).
  - Does *not* allow for a descriptive comment to be added to the backing store.
- Editing the `iw.cfg` file—Defines backing stores with entries in the [`iwserver`] section of the `iw.cfg` file.
  - Accepts multibyte characters for the store name (though the path to the store must use ASCII characters)
  - Creates the new backing store in any location.

- Allows you to add a descriptive comment to the backing store. This comment is displayed when the active backing stores are listed from the command line, or displayed in the TeamSite GUIs.
- Must be activated by using the `iwstoreadm` CLT with the `-a` option.

If you want to define backing stores by editing the `iw.cfg` file, complete the procedure described in the next section. If you want to create backing stores using the `iwstoreadm` CLT, complete the procedure described in “Creating Backing Stores Using the `iwstoreadm` CLT” on page 265.

## Defining Backing Stores in the `iw.cfg` File

As previously mentioned, the advantages of defining backing stores in the `iw.cfg` file include the ability to use multibyte characters in store names and to locate the backing store in a directory other than `/local/iw-store/`.

User-defined backing stores which are named using multibyte characters, must have a corresponding entry in the `iw.cfg` file. While the name of the backing store can be defined in multibyte characters, the backing store location *must* be defined using ASCII characters. All backing store data is stored in UTF-8 encoding.

Complete the following procedure to create backing stores defined in the `iw.cfg` file:

1. Ensure that you are logged in as the user root and that root has an entry in the Master role file (`iw-home/conf/roles/master.uid`).
2. Open the `iw.cfg` file in a text editor.

By default, the `iw.cfg` file is located in `/etc`.

3. If you are using multibyte characters for the store name, specify the encoding of your `iw.cfg` file by creating the following entry as the first line in the file—it *must* be the first line or it will be ignored.

```
[iwcfg]
encoding=locale_name
```

where `locale_name` is one of the following locales:

- iso-8859-1 (French or German)

- euc-jp (Japanese)
- shift-jis (Japanese)

For example:

```
[iwcfg]
encoding=shift-jis
```

**Note:** The locale entry must match the encoding of your text editor. Refer to page 324 for details about text editor encodings.

4. Append the following entry to the [iwserver] section to define additional backing stores:  
*store\_directory\_store\_name=absolute\_path\_to\_backing\_store*

For example:

```
store_directory_salesAsia=/local/salesAsia
```

**Note:** The *absolute\_path\_to\_backing\_store* must be in ASCII while the *store\_name* and the optional *descriptive\_comment* (described in step 5) can be in high-ASCII or multibyte characters.

5. Optionally, add a comment to the [iwserver] section below the backing store you just defined:  
*store\_comment\_store\_name=descriptive\_comment*

For example:

```
store_comment_salesAsia=Store for Demo
```

The completed entry, should look like this:

```
[iwserver]
existing iwserver entries
store_directory_salesAsia=/local/salesAsia
store_comment_salesAsia=Store for Demo
```

6. Save and close the `iw.cfg` file.
7. Run the `iwreset` CLT to have the TeamSite server read the changes to the `iw.cfg` file.

8. Run the `iwstoreadm` CLT with the `-a` option to create the newly defined backing store:

```
% iwstoreadm -a salesAsia
```

The `iwstoreadm` CLT checks the `iw.cfg` file to see if a `store_directory` or `store_comment` entry exists, when it finds these entries, their definitions are used to create the backing store.

The system then activates and mounts the new backing store.

9. Run the `iwstoreadm` CLT with the `-l` option to list all active backing stores:

```
%iwstoreadm -l
```

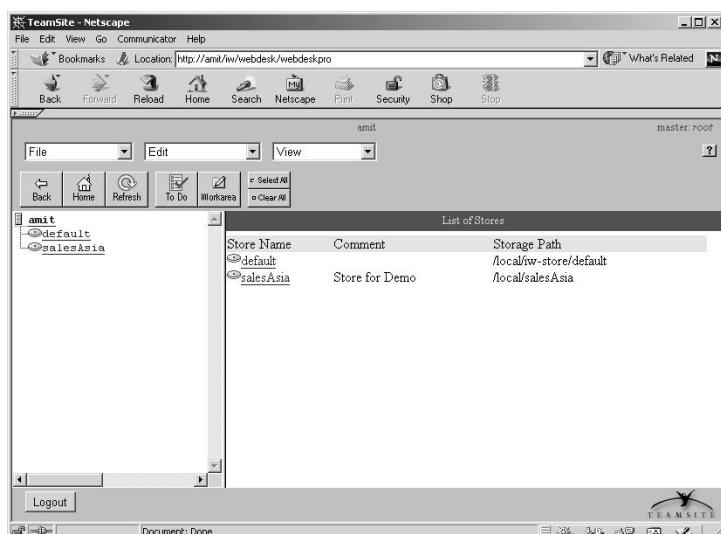
The system displays the following:

Name	Store Directory	ID	Comment
default	/local/iw-store/default	0x64	
salesAsia	/local/salesAsia	0x65	Store for Demo

10. Open your web browser and log in to WebDesk Pro or WebDesk.

11. Click **Workarea** (WebDesk Pro) or the **Files** tab (WebDesk).

The backing store and comment you created is listed in the GUI.



**Notes:**

- You can repeat the procedure to create any number of backing stores, but you can only have eight active at one time.
- In a MultiStore environment, you cannot relocate backing stores (`iw-store`) by editing the `/etc/defaultiwhome` file.
- You can edit the `store_directory_storename` entries to move backing stores defined in `iw.cfg`.

**Creating Backing Stores Using the iwstoreadm CLT**

The following procedure describes the creation of backing stores from the command line using `iwstoreadm`. It also describes viewing the newly created backing stores in both the command window and the TeamSite WebDesk Pro interface.

1. Ensure that you are logged in as the user root and that root has an entry in the Master role file (`iw-home/conf/roles/master.uid`).
2. Issue the `iwstoreadm -a store_name` command to create a new store, for example:

```
%iwstoreadm -a store1
```

`store1` is created in `/local/iw-store/`, mounted, and activated.

3. Type the following command to list the active backing stores:

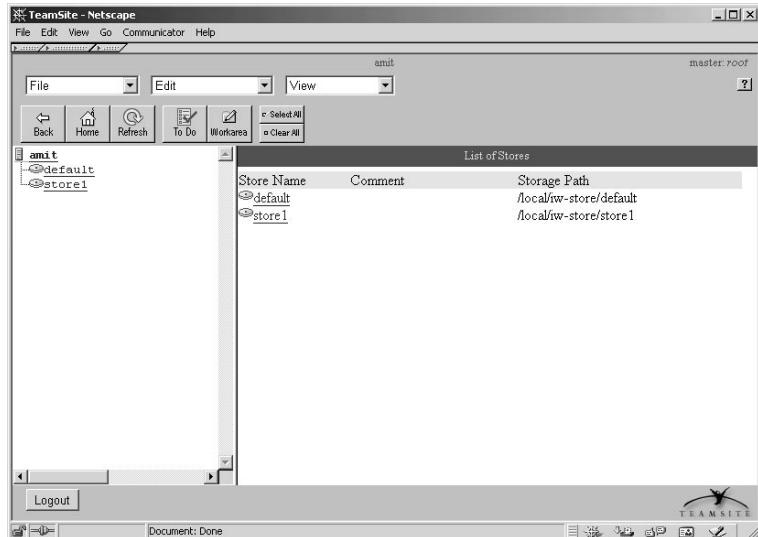
```
%iwstoreadm -l
```

The following listing is displayed:

Name	Store Directory	ID	Comment
default	/local/iw-store/default	0x64	
store1	/local/iw-store/store1	0x65	

4. Open your web browser and log in to WebDesk Pro or WebDesk.
5. Click **Workarea** (WebDesk Pro) or the **Files** tab (WebDesk).

The backing store you created is listed in the GUI.



## 6. Click **store1**.

Note that all backing stores (including the system-generated **default**) contain a **Main** branch, and a **STAGING** area that is based on an **INITIAL** edition.

# Administration CLTs

This section describes the new command-line administration utilities designed for use with the conversion GUI and new backing store functionality. The conversion CLT (`iwconvert`) is described on page 256.

## **iwstoreadm**

Backing store administration involves creating, activating, and deactivating backing stores by using the `iwstoreadm` CLT. When the `iwstoreadm -a storename` command is issued, the following sequence is triggered:

- The `iw.cfg` file is checked to see if a `store_directory` or `store_comment` entry exists, if it does, their definitions are used to create the backing store. If these entries do not exist:
- The backing store directory is automatically created in `/local/iw-store/` and populated with a minimal backing store containing a branch named `main`. The store remains active until explicitly deactivated by using `iwstoreadm -d` (even if the server is stopped and restarted).

Deactivating a store does *not* delete it. A store can be deactivated, moved to a new location, and reactivated using original name though it will be assigned a new store ID.

### **iwstoreadm Options**

The following options are valid for the `iwstoreadm` command:

<code>-a <i>store_name</i></code>	Activates an existing store, or creates and activates a new store. You must be logged in as the root user and root must have an entry in the Master role file ( <code>iw-home/conf/roles/master.uid</code> ) to execute this option.
<code>-d <i>store_name</i></code>	Deactivates an existing store. You must be logged in as the root user and root must have an entry in the Master role file ( <code>iw-home/conf/roles/master.uid</code> ) to execute this option.

- h                      Displays the usage message.
- l                      Lists active stores.

**Usage**

```
iwstoreadm [-l] [-a store_name] [-d store_name]
```

**Example**

```
%iwstoreadm -l
```

Displays the active backing stores:

Name	Store	DirectoryID	Comment
default	/local/iw-store/default0x64		
store2	/local/iw-store/store20x65		

**iwidmap**

A command line tool called **iwidmap** is included to change the mapping between the UID and the token. It can also be used to refresh the mapping when the same names are used, and the UID has changed. For more information about UID mapping, see “UID Changes to the TeamSite Backing Store” on page 274.

**Usage**

```
iwidmap [-v] [-h] (-u | -g) [-a][-c <user1> <user2>] [-x <file> | -i <file>] backing-store
```

- v                      Displays the version of this program.
- h                      Displays the usage message.
- u                      Update **userid** mapping.
- g                      Update **groupid** mapping.
- a                      Update all entries in the ID map.
- c <user> <user2>    Update user1 to user2.
- x <file>              Extract to file.

-i <file>	Import from file.
backing-store	Location of the backing store.

**Example**

```
iwidmap -u -c jgarcia rhunter /iw-store/NewReleases
```

## iwmigrate

The **iwmigrate** CLT is similar to **iwconvert** except that it accepts new-format backing stores as its source. It can be used to split a single new-format backing store into multiple backing stores, or to move the contents of a store to another location without losing the history of submit events for the editions.

Updates to the **iwconvert** and **iwmigrate** tools shipped with TeamSite 5.5.2L will be available on the Interwoven support website. Before using either CLT, check the Interwoven support website (<http://support.interwoven.com>) to ensure you have the most recent version of each tool.

### Usage

```
iwmigrate [-h] [-v] [-x] [-m mount_location] -o new_backing_location
[-b branch_vpath] [-s starting_ed] [-e ending_ed] [-n old_backing_location]
[-w workarea_name] [-c] [-d] [-f] [-l]
```

-h	Display this message.
-v	Display version number.
-b branch	Specify source branch for migration.
-x	Increase verbosity level. Maximum verbosity is level 3, expressed as -x -x -x.
-m mount_location	Specify mount location of backing store, for example: /iwmnt/default
-o new_backing_location	Specify new backing store location.
-n old_backing_location	Specify old backing store location.
-d	Do <i>not</i> recursively convert subbranches
-s starting_ed	Specify starting edition for migration. Default is the INITIAL edition (this option can only be used with -d)
-e ending_ed	Specify ending edition for migration. Default is the most recent edition.
-f	Full clone of every edition (does <i>not</i> preserve history).
-r	Clean up history information (use on the final pass).
-l locale	Specify the native locale of the backing store being migrated (if different from LC_LOCAL for this system).

**Example**

```
iwmigrate -m iwmnt/default/Safari -o /iw-store/safari_on_line -b
```

**iwconvertserver**

The `iwconvertserver` process supports the conversion GUI by communicating with remote TeamSite servers and invoking `iwconvert` on behalf of the GUI. You must run the `iwconvertserver` process manually for the GUI to function properly.

1. Change to the `iw-home/bin` directory, for example:

```
%cd /iw-home/bin
```

2. Run the `iwconvertserver` utility from `/iw-home/bin` with no arguments.

**iwcpfile**

The `iwcpfile` CLT copies a single file from a workarea on a pre-5.5 TeamSite server to a workarea on a TeamSite 5.5.2L server. This CLT enables you to copy files from workareas on your old (source) system that have not been submitted to STAGING prior to your conversion.

File history is *not* preserved for files copied with this CLT.

**Usage**

```
iwcpfile [-h][-v][-x] -m directory source_vpath dest_vpath
```

**-m *directory***      Specifies the location of the TeamSite file system corresponding to *source\_vpath*.

***source\_vpath***      Specifies the vpath to a file or symlink to be copied on the source server.

***dest\_vpath***      Specifies the target workarea to where the file is copied. If a file name is specified, the file is renamed. If the destination ends with the workarea name, the copied file retains the original name.

**-x**      Increments the verbosity level (may be specified more than once).

**-h**      Displays the usage message.

**-v**      Displays the version string.

## Example

From the TeamSite 5.5.1 server, mount the source (pre-5.5) TeamSite server. You must be logged in as root:

```
%mount -o vers=2 server_name:/iwserver /mount_location
%iwcpfile -m /mount_location //server_name/default/main/WORKAREA/
source_workarea/source_file /default/main/WORKAREA/target_workarea/
target_file
```

Root must also be an authorized TeamSite user.

## **iwcpwa**

The **iwcpwa** CLT copies all modified files and directories from a workarea on a pre-5.5 TeamSite server to a workarea on a TeamSite 5.5.2L server. This CLT enables you to copy multiple files from workareas on your old (source) system that have been modified, but not submitted to STAGING prior to your conversion.

### **Usage**

```
iwcpwa [-x][-v][-h][-d][-o] -m directory source_wa_vpath dest_wa_vpath
```

<b>-m <i>directory</i></b>	Specifies the location of the TeamSite filesystem corresponding to <i>source_wa_vpath</i> .
<i>source_wa_vpath</i>	Specifies the source workarea to copy modified files from.
<i>dest_wa_vpath</i>	Specifies the target workarea to copy the modified files to.
<b>-d</b>	Delete files in the output workarea as needed. By default this is turned off.
<b>-o</b>	Overwrite already modified files in the output workarea. By default this is turned off.
<b>-x</b>	Increments the verbosity level (may be specified more than once).
<b>-v</b>	Prints the version string.
<b>-h</b>	Prints the usage message.

## Example

From the TeamSite 5.5.1 server, mount the source (pre-5.5) TeamSite server. You must be logged in as root:

```
%mount -o vers=2 server_name:/iwservr /mount_location
%iwpwma -m /mount_location //server_name/default/main/WORKAREA/
source_workarea /default/main/WORKAREA/target_workarea
```

Root must also be an authorized TeamSite user.

## iwwfconvert

The `iwwfconvert` CLT copies active workflow jobs from your source backing store to your new backing store. Workflow conversion is the last step of the backing store conversion process and can only be performed after all the editions and workareas required by the workflow are recreated on the target server.

The `iwwfconvert` CLT first checks to ensure all the vpaths referred to by the workflow objects exist in the new backing store. If any of the vpaths are *not* found, `iwwfconvert` displays an error and stops. If all the vpaths are found, `iwwfconvert` converts the workflow objects.

### Notes:

- Before starting the workflow conversion, run `iwfreeze` to freeze your old backing store and `iwstoreadm -d` to deactivate your new backing store.
- Once the workflow conversion is done, any default workflow backing store created by the new teamsite server should be overwritten by the new converted store.
- The new backing store structure must match the old backing stores structure. That is, `/default/main/branch1` is converted as `/default/main/branch1` (multiple stores cannot be used)

## Usage

```
iwwfconvert -o output_workflow_backing_store -i input_workflow_backing_store -n
new_fs_backing_store [-h] [-v]
```

-o	Location of the output workflow backing store.
-i	Location of the input workflow backing store.
-n	Location of the new file system backing store.
-c	Confirms that the workflow conversion can proceed (does not start the actual conversion).
-h	Display help message.
-v	Display version number.

## Example

```
iwwfconvert -o target/local/iwstore/workflow -i source/local/iwstore/
workflow -n target/local/iwstore/default
```

## UID Changes to the TeamSite Backing Store

Previous releases of TeamSite have stored UNIX user IDs (UID) and group IDs (GID) representing file ownerships and access control directly in the backing store. This would cause problems when converting the backing store onto different systems.

The new-format backing store uses a unique 32-bit ID generated by the TeamSite server instead of storing the UID. A one-to-one persistent mapping exists between the TeamSite generated ID and the UID. Whenever the UID has to be written out to the backing store, the mapping is checked to obtain the TeamSite ID, which is substituted for the UID. When a restore is attempted, the reverse lookup takes place and the appropriate UID is recovered.

A command-line tool called *iwidmap* is included to change the mapping between the UID and the token. It can also be used to refresh the mapping when the same names are used, but the UID has changed. Details about the *iwidmap* CLT are included on page 268.

## Chapter 9

# Backing Up TeamSite

---

Your TeamSite backing stores represent a tremendous investment in resources and are a valuable corporate asset. As such, they should be backed up daily, or even more frequently, to minimize the possibility of damaged or lost data. TeamSite 5.5.2 requires the use of third-party backup solutions. Any backup mechanism that can guarantee exact time and state directory content recovery can be used.

The command-line tools for backing up and restoring TeamSite are no longer included or supported.

## Integrating with Third-Party Backup Solutions

Interwoven recommends using a high quality third party backup solution for protecting the backing store data. When evaluating a backup solution, the following criteria are essential:

- The backup method must provide a way to perform an `iwfreeze` operation prior to performing the backup. This must be done to assure that the backing store does not change during the backup. The backup method must then perform an `iwfreeze --` operation to allow writes to the backing store when the backup is finished.
- The backup method must be fast enough to perform a full or incremental backup of the backing store within a reasonable length of time. The maximum allowable length of time depends on the requirements of the particular installation, but should probably be less than 12 hours.
- The restore method must provide a way to do a complete state-restore of a directory as of a given time. This means that when a directory is recovered, the contents must match exactly what was in the directory at the time the backup was performed. Only files that were present at the time of the backup must be present in the restore. That is, if a file was deleted from the original directory between backups, it should not be present in the restore. Some backup and restore products regard all backed-up files to be “sticky,” that is, as long as a file ever existed, it will be present in the restoration regardless of whether it was deleted prior to the last backup.

Additional criteria to consider are:

- An automated backup execution facility capable of performing full backups followed by level (preferred) or incremental backups to provide a customizable backup strategy.
- Automated backup media management and manipulation (for example, a tape jukebox or silo).
- The ability to make copies of completed backups for offsite storage.

If the available backup method is efficient and inexpensive (compared to the value of the data being protected), the TeamSite workareas can also be backed up to allow users to recover individual files or directories from their workareas, rather than having to recover the entire backing store. This is a very convenient feature for users, but can come at a relatively high price in terms of extra time and space needed for these redundant backups. Although the virtual files which comprise much of TeamSite's file system mount (`/iwmnt`) take up no extra space on the TeamSite server, if the actual TeamSite workareas are backed up, the virtual files in the workareas will be treated as actual files and will take up space in the backup media.

You must freeze the TeamSite backing store (with the `iwfreeze` command) while you are backing up your backing stores or workareas. Failure to freeze the backing store while you are backing up can result in possible data loss and corruption. For details about the `iwfreeze` CLT, refer to the *Command-Line Tools* manual for your platform.

If you are using multiple backing stores, you can back up each store independently. The `iw-store` directory should be backed up if you have in-progress workflows or batch jobs that you do not want to lose. You can freeze and unfreeze the workflow store just like any other store, but you cannot move it outside of `iw-store`.

Backing up workareas alone is not a substitute for backing up the TeamSite backing store. If you only back up the files that appear in the TeamSite file system mount, you will lose important metadata such as version histories and file status. Always back up the actual TeamSite backing store whether or not you back up individual workareas.

## Suggested Strategies for Incremental Backups

It is possible to implement a “level-oriented” backup if a sufficiently sophisticated backup solution is available. For example, a full backup can be performed on the first Saturday of the month, then incremental backups that build on each other can be performed for the rest of the week. On the second Saturday of the month, a “super-incremental” backup based on the original full backup done on the first Saturday is performed. The super-incremental backup supersedes all of the previous incremental backups. Only the first full backup and super-incremental are needed to completely recover the backing store. For the subsequent week, incremental backups are again performed based on the super-incremental backup done on the second Saturday. The following Saturday, another super-incremental backup based on the previous super-incremental file is performed, again eliminating the need for the previous week’s incrementals to recreate the backing store. To perform a recovery at this point, restore the original full backup, then each super-incremental in sequence, and finally the balance (if any) of the current week’s incrementals.

This tiered, or level-oriented backup can be repeated on a monthly basis to produce a week-by-week record of the backing store. To reproduce the backing store as of any particular Saturday, recover the full backup from the beginning of the month, then apply each Saturday backup in turn until the desired Saturday is reached.

To determine your optimal backup strategy, you must analyze the trade-offs of convenience and speed in backing up versus simplicity and speed of restoration, and decide what best suits your needs. A strategy using a single full backup and an indefinite string of incrementals is optimized for backup speed, but the amount of time required to perform a full recover of the backing store grows with each passing day as a new incremental is added to the list. Every backup must be preserved to be able to recover the backing store. One benefit of this method is that a complete daily record of the backing store will be preserved.

The opposite extreme is to perform a full backup every day. Each backup will take the maximum amount of time to perform, but only one recover needs to be done to completely recreate the backing store. If you only preserve the previous day’s backup, no history of the backing store will be retained, but the amount of storage space used by the backups is minimized.



## Appendix A

# TeamSite Configuration Files

The following files contain information about your TeamSite server configuration:

Configuration File	Function
/etc/defaultiwhome	Describes the location of the TeamSite application software. The installation default value is /usr/iw-home.
/etc/defaultiwstore	Describes the location of the TeamSite server backing store directory. The installation default value is /usr/iw-store.
/etc/defaultiwmount	Describes the location of the TeamSite virtual mount point. The installation default value is /iwmnt.
/etc/defaultiwlog	Describes the location of the iwserv.log file. The installation default value is /var/adm/iwserv.log.
/etc/defaultiwevlog	Describes the location of the iwevents.log file. The installation default value is /var/adm/iwevents.log.
/etc/defaultiwtrace	Describes the location of the iwtrace.log file. The installation default value is /var/adm/iwtrace.log.
iw-home/etc/iw.cfg (default location—see “Location of iw.cfg” on page 281 for more information)	Contains various parameters necessary for the operation of TeamSite, as described in the Chapter 5, “Configuring the TeamSite Server.”
iw-home/iw-samba/lib/iw.smb.conf	Contains Samba configuration.

<b>Configuration File</b>	<b>Function</b>
<code>iw-home/local/config/submit.cfg</code>	Specifies all file permissions that will automatically be changed at submit time.
<code>iw-home/local/config/autoprivate.cfg</code>	Specifies what types of files will automatically be marked private.
<code>iw-home/local/config/file_encoding.cfg</code>	Contains rules that determine the character encoding of the contents of files that do not specify their encoding. See page 283 for information about creating these rules.
<code>iw-home/local/config/iwtemplates.cfg</code>	Specifies which New File templates will be used in which TeamSite areas.
<code>iw-home/conf/roles/master.uid</code>	Contains a list of all users who can log in as a Master user.
<code>iw-home/conf/roles/admin.uid</code>	Contains a list of all users who can log in as an Administrator.
<code>iw-home/conf/roles/editor.uid</code>	Contains a list of all users who can log in as an Editor.
<code>iw-home/conf/roles/author.uid</code>	Contains a list of all users who can log in as an Author.

The locations of most of these files can be changed (see See “File Locations” on page 156.).

## Location of iw.cfg

If `iw.cfg` does not exist in the default location, TeamSite will look for it in the following locations, in order:

`/etc/iw.cfg`

`iw-home/config/iw.cfg`

`iw-home/local/etc/iw.cfg`

`iw-home/etc/iw.cfg`

If `iw.cfg` is not found in any of these places, TeamSite will assume the default values for `iw.cfg` settings.

## Location of Roles Files

TeamSite looks for the roles files (`author.uid`, `editor.uid`, `admin.uid`, `master.uid`) in the following locations, in order:

The value in the `iwroles` field in the `[locations]` section of `iw.cfg`, if it exists.

`iw-home/conf/roles`

`iw-home/local/config/roles`

`iw-home/config/roles`

`iw-home/local/config/roles`



## Appendix B

# Specifying Content Encoding

---

TeamSite now includes a configuration file called `file_encoding.cfg` that enables you to create rules to determine the character encoding of the contents of files that do not specify their encoding. The `file_encoding.cfg` file (located by default in `iw-home/local/config`) uses an XML-based language called `regex_map`. The `regex_map` format is designed to be structured enough for maintainability, and extensible so that the same format may be used in future configuration files. This file contains a `<regex_map>` element, which contains rules to map `vpaths` (directory paths) to the character encoding specification of file contents.

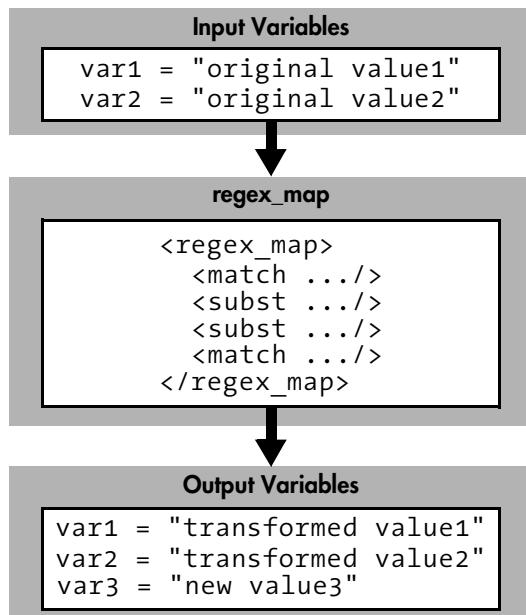
For TeamSite to correctly interpret a text document, it is necessary to know the character encoding in which its contents are represented. Unlike an HTML document that can declare the encoding of its contents using an `<HTTP META="Content-Type" CONTENT="text/html; charset=charsetname">` tag, a plain text file has no mechanism for storing this information. The encoding is required for certain TeamSite functionality including SmartContext Editing (SCE) and the Source Differencing and Interwoven Merge tools.

In previous releases, SCE relied on the Content-Type header from the content webserver to specify the encoding of plain text files. This required you to configure the encoding at your content webserver which may limit flexibility and scalability. By default, the Source Differencing and Interwoven Merge tools assumed that text files are encoded in ISO-8859-1, which is not suitable for content in eastern Asian languages.

The sections that follow describe the `regex_map` language, and how it is used to specify the character encodings of text files used by TeamSite.

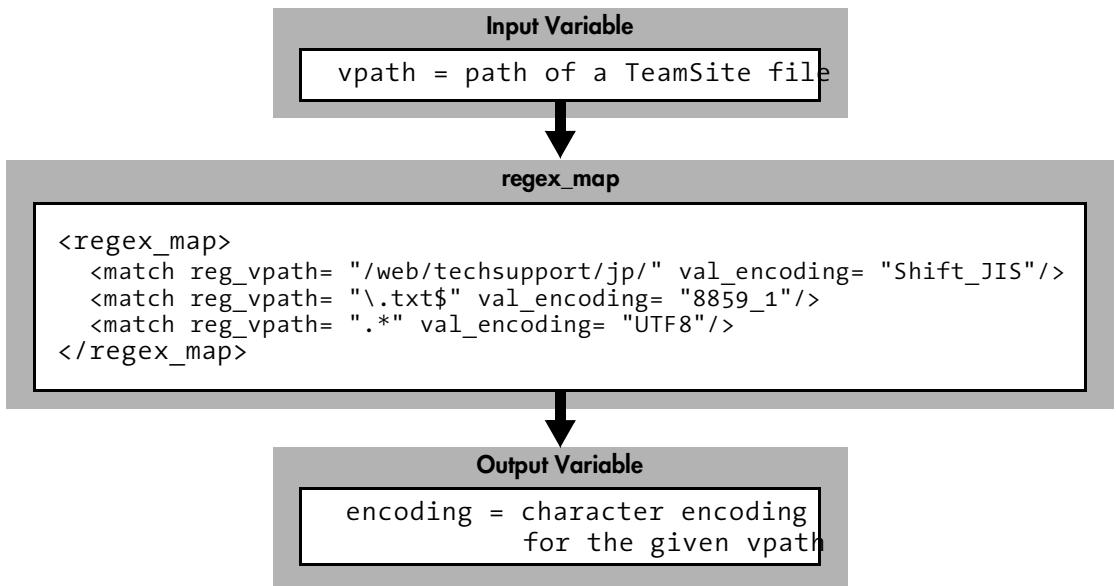
## regex\_map Defined

A `regex_map` is a filter that transforms a set of input variable values into a set of output variable values through a set of rules written in XML using the following form:



## Simple regex\_map Example

The following `regex_map` determines the character encoding of TeamSite files. Each `reg_vpath` means that a match is to be performed on the `vpath` variable, and each `val_encoding` assigns a result if the match succeeded.



In the preceding `regex_map` example:

- If the input `vpath` variable is `"/x/y/z.txt"`, the resulting `encoding` variable is set to `"8859_1"` because `"/x/y/z.txt"` ends with `".txt"`.
- All files in the `/web/techsupport/jp` branch are encoded in Shift-JIS, because their `vpath` begins with `"/web/techsupport/jp/"`.
- If the input `vpath` variable is anything other than `"/web/techsupport/jp/"`, the output `encoding` variable is set to `"UTF8"` because `".*"` matches any string.

Note that each rule within `<regex_map>` is evaluated in order, and that the first `<match>` tag with a regular expression that matches the input variable `vpath` is used and subsequent rules are ignored. Therefore, it is important for the `<match reg_vpath= ".*" val_encoding= "UTF8"/>` rule to appear last.

## The `regex_map` Format

A `regex_map` consists of a `<regex_map>` element that contains substitution and match rules expressed by using `<subst>` and `<match>` tags. Substitution and match rules are consulted in the order in which they are listed within the `<regex_map>` element. Each rule may assign values to variables.

### Rule Syntax

Every `<subst>` or `<match>` rule expresses the following logical operation:

If all the regular expressions within this rule match, then perform all of this rule's variable assignments; otherwise, ignore this rule and consult the next rule.

Execution terminates when the first `<match>` rule has been applied, or when there are no more rules. A `<subst>` rule that has been satisfied does not terminate execution (unless it is the last rule).

All attributes of rules use the form `reg_varname` or `val_varname`.

- `reg_varname` attribute—Applies a regular expression to `varname`.
- `val_varname` attribute—Assigns a value to `varname` if all of the regular expressions in the current rule are satisfied.

The following are some simple examples of rules.

- If `vpath` starts with `"/default/main/"` set the encoding to `"8859_1"` and continue processing:  
`<subst reg_vpath="^/default/main/" val_encoding="8859_1"/>`
- The encoding of all files named `"index_zh_TW.html"` anywhere in the `/web` branch is `"Big5"`. There are no exceptions to this rule, so stop processing if it applies.  
`<match reg_vpath= "^\web/(STAGING|WORKAREA|EDITION).*/index_zh_TW.html" val_encoding= "Big5"/>`

Note that the “or” capability of regular expressions, expressed by the pipe character (`|`), enables this single rule handle three cases at once (STAGING or WORKAREA or EDITION).

- The encoding is always "Shift\_JIS".

```
<match val_encoding="Shift_JIS"/>
```

When there are no `reg_` conditions, the assignment always executes if the rule is encountered. Any rules that occur after this statement are unused.

## Regular Expression Syntax

The `regex_map` interpreter uses Perl-Compatible Regular Expressions (PCRE) as its regular expression engine. The PCRE is similar to the Perl regular expression engine and includes advanced features such as lookahead assertions.

Regular expressions in `regex_map` are case-sensitive by default. If a variable is listed in the `opt_case_insensitive` attribute of `<regex_map>`, all regular expressions applied to that variable in the `regex_map` are case-insensitive.

For example, because filenames and URLs are case-insensitive on Microsoft Windows, the following declaration would be recommended when writing a `regex_map` for a TeamSite server on Microsoft Windows:

```
<regex_map opt_case_insensitive="vpath url">
 <subst reg_vpath="..." .../>
 <match reg_url="..." .../>
</regex_map>
```

## Variables

Variables store strings to be passed in the following ways:

- as input to a `regex_map` from an application
- from rule-to-rule within a `regex_map`
- as results from a `regex_map` to the application

Variable names are case-sensitive and must begin with a letter and may contain any sequence of alphanumeric characters and the underscore character ("\_"). References to any variable whose value is not set by the application or by rules in the `regex_map` evaluates to an empty string.

## Application Variables

Any application that uses a `regex_map` gives it a set of inputs before execution and inspects a set of output variables when the `regex_map` processing completes. These input and output variables are known as **application variables**.

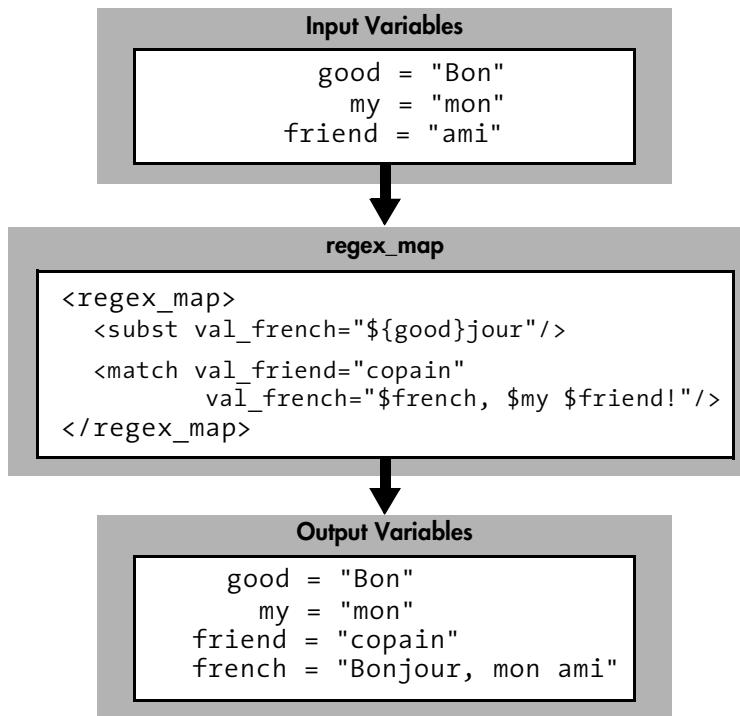
## Intermediate Variables

You may find it helpful to assign intermediate results to your variables in `regex_map` rules before arriving at final output values. These **intermediate variables** can help make a complex set of rules more manageable by factoring out several separate conditions into one condensed case. You can then write one rule to act on the condensed case, instead of repetitively writing the same actions for the individual initial conditions. “Strategies for Effective `regex_maps`” on page 294 contains an example of factoring.

Intermediate variables should have names that begin with `x_` to avoid conflicts with application variables that Interwoven may create in the future.

## Interpolation of Variables and Captured Subexpressions

When assigning a value to a variable, the values of other variables can be included. In `val_ attributes`, each occurrence of  `${varname}` or `$varname` causes the value of `varname` to be inserted instead, as shown in the following example:



In the preceding example:

- In the `<subst>` rule, curly braces (`{ }`) are required to separate the variable name `good` from the literal string `jour` that immediately follows.
- In the `<match>` rule, there is no need to disambiguate the three variables because the variable names `french`, `my`, and `friend` are followed by a comma, a space, and an exclamation point, none of which can be confused as being part of a variable name.
- In the second rule, the values of `friend` and `french` are taken from the time at which the rule was encountered. All assignments in a rule appear to occur simultaneously and do not affect each other.

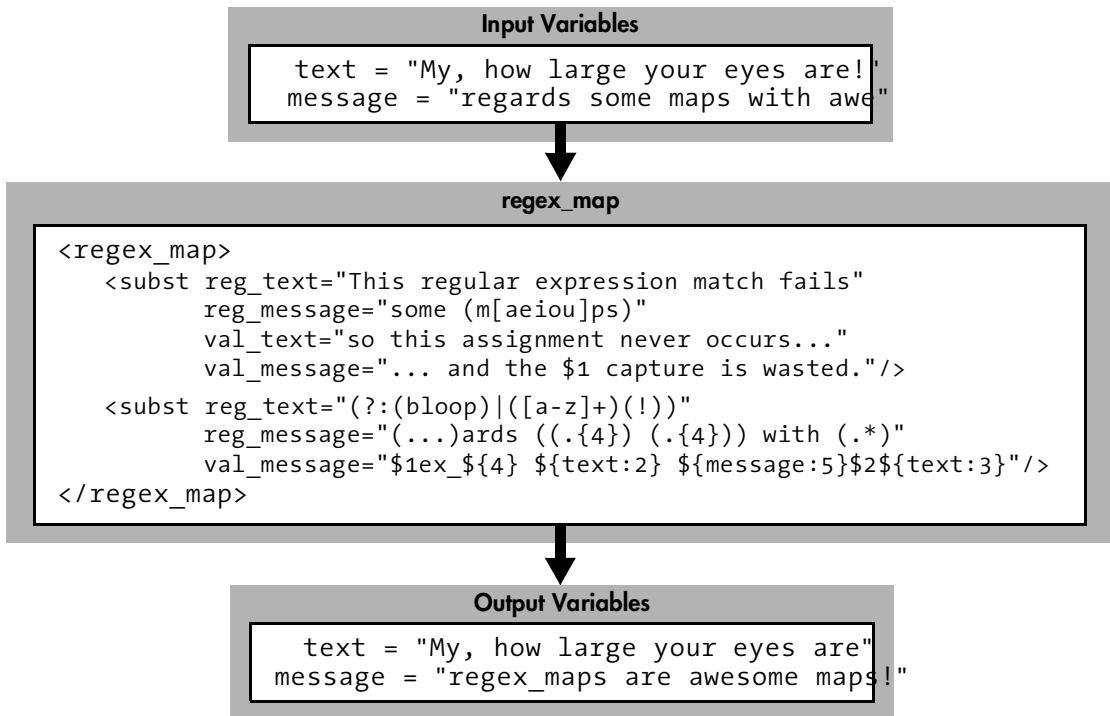
It is also possible to include just portions of variables. Placing parentheses around portions of a regular expression applied to `varname` creates a **captured subexpression** variable that can be used when assigning values. The longhand form of a captured subexpression variable is  `${varname}:n}`, which refers to the string captured by the  $n^{\text{th}}$  pair of parentheses in `reg_varname`.

**Note:** Pairs of parentheses are numbered according to the order in which their left parenthesis occurs within the regular expression. Parentheses of the form `(?:some_expression)` are used solely for grouping characters in the regular expression, not for capturing text during matching, and are excluded from the count.

The shorthand version of a captured subexpression variables is `$n`. Note that the shorthand notation can only be used when the variable being modified is the same as the variable from which the subexpression was captured.

Unlike application and intermediate variables, captured subexpression variables are scoped to the `<subst>` or `<match>` rule that created them. If a captured subexpression variable needs to be used in a subsequent rule, it should be stored in an intermediate variable.

For example, the pair of rules in the following `regex_map` makes the assignment `message="regex_maps are awesome maps!"` in an inefficient way:



In the previous example:

- The first rule does not apply because the value of the `text` variable does not match the regular expression in `reg_text`.
- While performing the regular expression match for `message`, the special variable  `${message:1}` (the `$1` variable associated with `message`) takes on the value `maps` within the scope of the rule. However, since the entire rule is inapplicable, it has no effect. Neither of the two `val_assignments` happens, and the temporary  `${message:1}="maps"` binding is discarded.

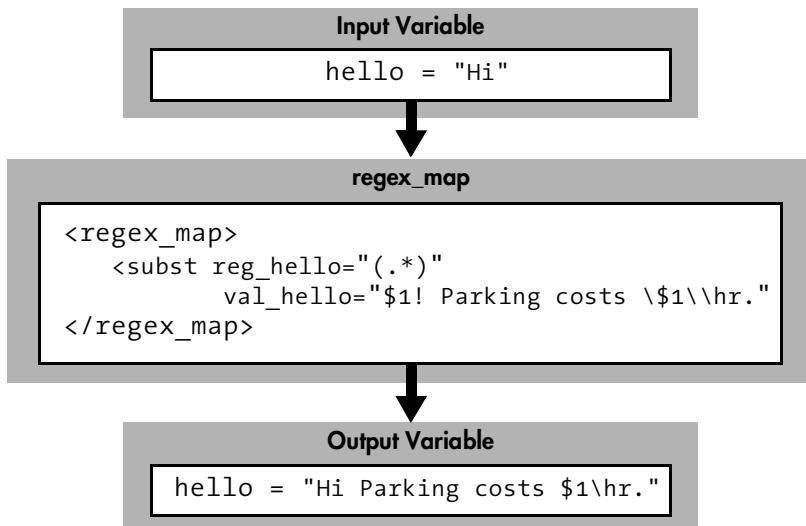
- The second rule does apply, since both of the `reg_text` and `reg_message` matches succeed. The parentheses also capture the text in the strings, resulting in the following temporary bindings:

```
 ${text:1} = empty string
 ${text:2} = are
 ${text:3} = !
 ${message:1} = reg
 ${message:2} = some maps
 ${message:3} = some
 ${message:4} = maps
 ${message:5} = awe
```

- Finally, variable interpolation occurs for the `val_message` assignment. Since the `$1`, `${4}`, and `$2` occur in a `val_message` context, they are treated as shorthand for  `${message:1}`,  `${message:4}`, and  `${message:2}`, respectively. The curly braces for  `${4}` are optional in this case, and could be used in other situations to clarify where the variable name ends and literal text begins.

## Quoting

Inside a `val_` attribute, dollar signs (\$) have special meaning—they mark the start of captured subexpression names. To force a dollar sign to lose this special meaning (and be treated as a literal dollar sign), it must be escaped by preceding it with a backslash. Similarly, a backslash is treated as a special quoting character unless it is escaped by a preceding backslash.



In the preceding example, `hello` is assigned the value "Hi! Parking costs \$1\hr." (with the deliberately "wrong" backslash used instead of a forward slash for demonstration purposes).

Also, because `regex_map` is written in XML, characters with special meaning in XML need to be represented using XML entities. These special characters are described in the following table.

Special XML Character	Visual Representation	XML Entry
Double quote	"	&quot;
Apostrophe	'	&apos;
Ampersand	&	&amp;
Greater than	>	&gt;
Less than	<	&lt;

For example,

```
<subst val_statement="Programmers think "1 & 1 is 1."" />
```

assigns the following value to the `statement` variable:

`Programmers think "1 & 1 is 1."`

## Strategies for Effective regex\_maps

The `regex_map` grammar is a powerful string manipulation language yet still allows simple configurations to be expressed simply. This is due to:

- the ability to work with multiple variables
- the use of regular expressions with the capability to reference captured subexpressions
- the option to chain rules with `<subst>` or stop processing with `<match>`

By taking advantage of these features, you can write configuration files that are compact and manageable.

The following example demonstrates how factoring and intermediate variables can make a `regex_map` configuration scale to handle complex situations. Suppose that a system-wide reorganization forced you to rename all files named `README` to `README.TXT` and relocate all files under the `/a/b` branch to the `/c/d` branch. You could list all of the possibilities as follows:

```
<regex_map>
 <!-- Handle both branch move and file extension addition -->
 <match reg_vpath= "/a/b/((WORKAREA|EDITION|STAGING).*)README$"
 val_vpath= "/c/d/$1README.TXT"/>

 <!-- Handle branch move only -->
 <match reg_vpath= "/a/b/((WORKAREA|EDITION|STAGING).*)"
 val_vpath= "/c/d/$1"/>

 <!-- Handle file extension addition only -->
 <match reg_vpath= "(.*)/README$"
 val_vpath= "$1/README.TXT"/>
</regex_map>
```

But this strategy could become extremely complicated if there were more combinations. A factored set of rules can handle each change independently:

```
<regex_map>
 <!-- Handle a possible branch move -->
 <subst reg_vpath= "/a/b/((WORKAREA|EDITION|STAGING).*)"
 val_vpath= "/c/d/$1"/>

 <!-- Handle a possible file extension addition -->
 <subst reg_vpath= "(.*)/README$"
 val_vpath= "$1/README.TXT"/>
</regex_map>
```

A complicated set of rules could be clarified with intermediate variables, for example:

```
<regex_map>
 <!-- Decompose vpath into branch, area, directory, filename -->
 <!-- Decomposition could be done in just one rule, -->
 <!-- but we choose to break it up with the help of x_rest. -->
 <subst reg_vpath="^(.*)/((?:WORKAREA|EDITION|STAGING).*)"
 val_x_branch="${vpath:1}"
 val_x_rest="${vpath:2}"/>
 <subst reg_x_rest="((?:WORKAREA|EDITION)/[^/]+|STAGING)(.*)"
 val_x_area="${x_rest:1}"
 val_x_rest="${x_rest:2}"/>
 <subst reg_x_rest="(.*)(/.*)"
 val_x_dir="${x_rest:1}"
 val_x_file="${x_rest:2}"/>
 <!-- End decomposition -->

 <!-- Do the transformations -->
 <subst reg_x_branch="^/a/b$"
 val_x_branch="/c/d"/>

 <subst reg_x_file="^/README$"
 val_x_file="/README.TXT"/>
 <!-- End transformations -->

 <!-- Put vpath back together. -->
 <subst val_vpath="x_branchx_areax_dirx_file"/>
</regex_map>
```

In the preceding example, factoring out the vpath decomposition logic simplifies the actual transformation rules. In a complex situation with many transformation rules, adding a few standardized rules at the beginning and end of the `regex_map` is worthwhile.

“Advanced `regex_map` Example” on page 299 demonstrates of the expressiveness of `regex_maps` by showing how a Roman numeral can be incremented with sequential string substitution rules.

## Internationalization and `regex_map`

TeamSite `regex_maps` should be written in UTF-8 and should provide UTF-8 input values and expect UTF-8 output values for all variables. The regular expression engine is UTF-8-aware. For example, a period (.) in a regular expression matches a single character, regardless of the number of bytes needed to represent that character.

**Note:** If you need to specify non-ASCII literal characters in your `regex_maps`, ensure the text editor you are using can edit and save the `file_encoding.cfg` in UTF-8 encoding. Refer to page 322 for details about text editor encodings.

## SmartContext Editing and `file_encoding.cfg`

To determine the encoding of a text file, SmartContext Editing mimics the behavior of your web browser by performing the following series of checks:

- First, SCE checks the Content-Type header from the content webserver. If the MIME type (`text/html` or `text/plain`) is followed by a character encoding declaration (for example, `Content-Type: text/plain; charset=UTF-8`), it uses the specified encoding.
- If the file is an HTML document, SCE searches for a character encoding declaration in an HTML META tag (for example, `<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=Big5">`), which it uses if found.
- If the aforementioned methods do not return the encoding, SCE computes the encoding using the `regex_map` configuration in the `file_encoding.cfg` file. It uses the vpath as the input variable and the encoding as the expected output.

## Source Differencing and Merging and file\_encoding.cfg

Unlike SCE, the Source Differencing and Interwoven Merge tools do not mimic your web browser. Instead, they rely entirely on the `file_encoding.cfg` file to determine the character encoding of text documents. The Source Differencing and Interwoven Merge tools assume that the “other” file and the “common ancestor” file share the character encoding of the workarea file.

The following list of encodings are the IANA preferred charset names (<http://www.iana.org/assignments/character-sets>) and are valid entries for the `file_encoding.cfg` file:

English, French, German:

- ISO-8859-1
- ISO-8859-15
- windows-1252

Japanese:

- Shift\_JIS
- EUC-JP

Unicode:

- UTF-8

**Note:** `file_encoding.cfg` has no effect on the file encoding seen in visual differencing. This is so that what is seen in visual differencing tool most closely approximates what will be seen in the production environment.

## Sample file\_encoding.cfg

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Input variable: vpath -->
<!-- Output variable: encoding -->
<vpath_to_encoding_map>

<!-- Ignore upper and lower case when
 evaluating reg_vpath and reg_encoding conditions. -->
<regex_map opt_case_insensitive="vpath encoding">
 <!-- Set the default result. A default like this is highly recommended. -->
 <subst val_encoding="8859_1"/>

 <!-- Make a note of Japanese files scattered about. -->
 <subst reg_vpath="(?:_ja|_jp|_jpn)\."
 val_x_lang="Asian:Japanese"/>

 <!-- Likewise with Chinese files. -->
 <subst reg_vpath=".*\zh\.txt$"
 val_x_lang="Asian:Chinese"/>

 <!-- As site policy, our Japanese files are Shift-JIS -->
 <subst reg_x_lang="Asian:Japanese"
 val_encoding="Shift-JIS"/>

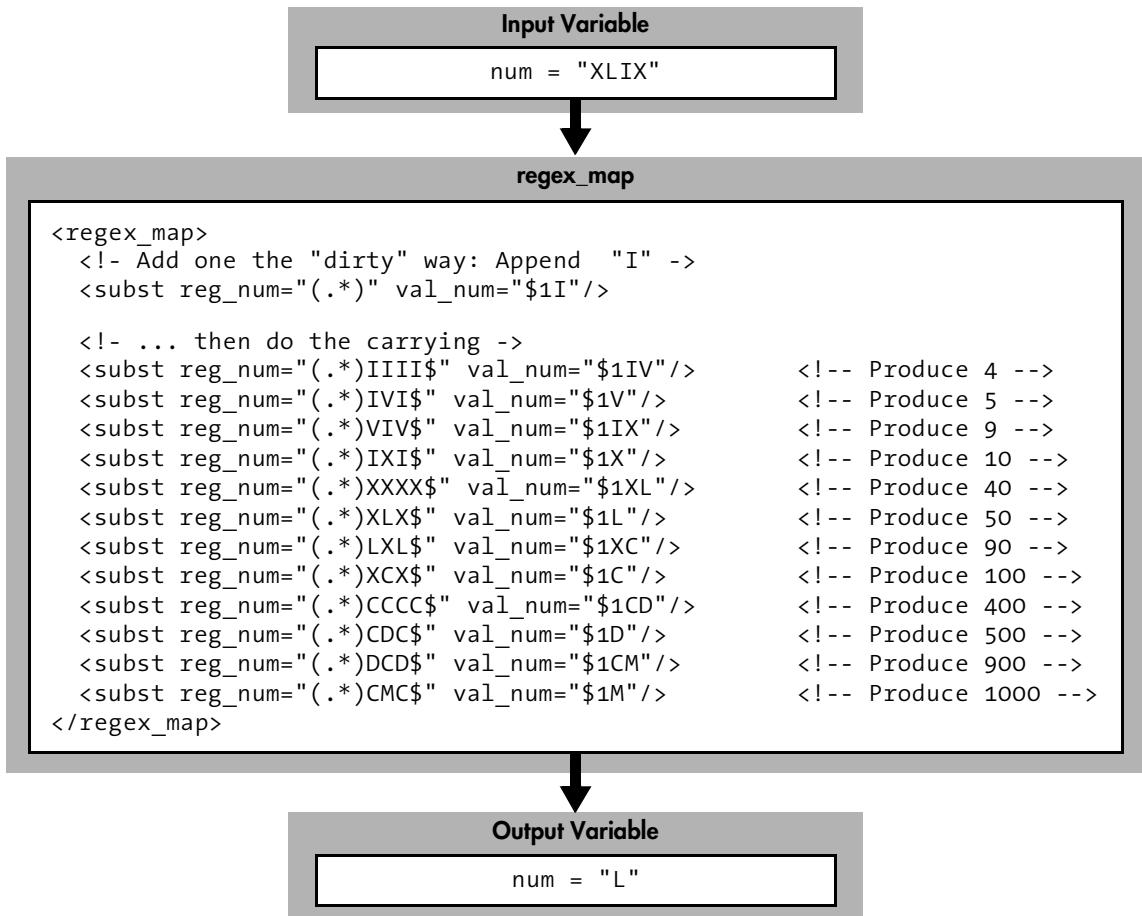
 <!-- As site policy, our Chinese files are Big5 -->
 <subst reg_x_lang="Asian:Chinese"
 val_encoding="Big5"/>

 <!-- Otherwise, the directory name at the top of the area is the result. -->
 <subst reg_vpath="(?:(:WORKAREA|EDITION)/[^/]+|STAGING)/([^\/]+)/"
 val_encoding="${vpath:1}"/>

 <!-- Canonicalize encoding names. Try Shift_JIS, then SJIS. -->
 <match reg_encoding="(sjis|shift[-]jis)"
 val_encoding="Shift_JIS"/>
</regex_map>
</vpath_to_encoding_map>
```

## Advanced regex\_map Example

This example `regex_map`, which adds one to a Roman numeral, demonstrates the power of chained substitution rules. It uses `num` as both the input and the output variable. The example below shows 49 being transformed into 50.



The `regex_map` language works well with Roman numerals because it is designed for string manipulation. It would be much more difficult to write a `regex_map` that increments Arabic numerals, due to the larger set of rules needed to increment a single digit and the lack of looping capability to perform the carrying operation.



## Appendix C

# High Availability TeamSite

---

This appendix describes TeamSite HA (High Availability). TeamSite HA has two aspects

- A watchdog daemon that monitors the TeamSite server.
- A “hot standby” integration with Sun Cluster 2.2 (Solaris only).

You must purchase TeamSite HA in addition to the base version of TeamSite to use the features described in this appendix.

## HA Watchdog

### About HA Watchdog

HA Watchdog uses a watchdog daemon and a set of associated tools and scripts to monitor the TeamSite server, detect process and power failures, log failure events, and optionally take corrective action. After TeamSite HA is installed and configured, it is transparent to end users, performing all user-visible operations in a way that is identical to the base version of TeamSite.

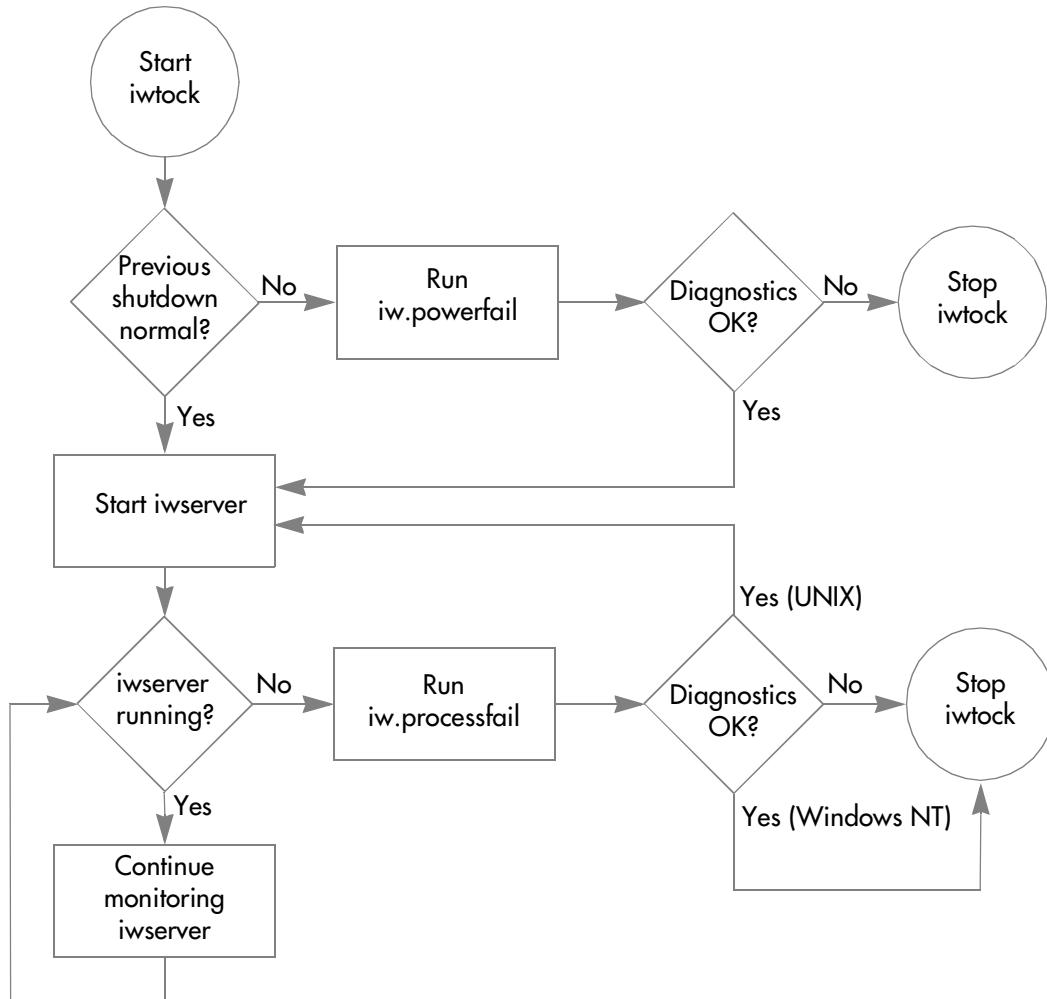
You can configure the HA Watchdog in several ways. For example, you can instruct it to:

- Shut down the TeamSite server and notify a specified system user that a power or process failure was detected.
- Stop the TeamSite server after detecting a failure even if all post-failure system checks are normal.
- Perform a different action depending on whether a power failure or a process failure was detected.
- Perform any other Perl script-defined action automatically upon failure detection.

You can also turn off the availability module completely, in which case the base version of TeamSite continues to run.

## TeamSite HA Watchdog Components and Processes

The following flowchart shows the TeamSite HA Watchdog components and processes. See the section immediately following the flowchart for an explanation of each item.



The main TeamSite HA Watchdog component is the `iwtock` watchdog daemon, which starts the `iwserver` process and tracks `iwserver` execution for as long as TeamSite server is running. When `iwtock` first starts, it determines whether the previous TeamSite shutdown was abnormal or normal. If it detects an abnormal shutdown, `iwtock` runs the `iw-home/ha/conf/iw.powerfail` script, which can be configured either to stop `iwtock` or to perform a variety of system checks or other actions as described later in the “Configuring TeamSite HA Watchdog” section. If the system meets the passing criteria defined in `iw.powerfail`, `iwtock` starts `iwserver`. If the system does not meet the passing criteria, `iwtock` stops and `iwserver` is not started. All output from `iw.powerfail` is logged in `iwserver.log`.

If `iwtock` determines that the previous TeamSite shutdown was normal, it starts `iwserver`. From this point on, `iwtock` continues to monitor `iwserver`. If at any time `iwtock` detects that `iwserver` is not running and there is no evidence of an explicit shutdown, it assumes that an unexpected shutdown or system interruption has occurred. In this situation, `iwtock` runs the `iw-home/ha/conf/iw.processfail` script, which can be configured either to stop `iwtock` or to perform a variety of system checks or other actions as described in “Configuring TeamSite HA Watchdog” on page 304. If the system meets the passing criteria defined in `iw.processfail`, `iwtock` starts `iwserver`. If the system does not meet one or more passing criteria, `iwtock` stops and `iwserver` is not restarted.

All output from `iwtock`, `iw.powerfail`, and `iw.processfail` is logged in `iwserver.log`.

If `iwserver` attempts to spawn more than once within 30 seconds of initial startup or a restart, `iwtock` will exit.

## Installing TeamSite HA Watchdog

Perform the following steps to install TeamSite HA Watchdog. After the installation is complete, you have access to the failsafe and availability modules.

1. Log in as **root**.
2. Stop `iwserver`:

```
% /etc/init.d/iwserver stop
```

3. Copy the TeamSite HA installation file—`IWOVha.5.5.2.BuildXXXX.tar.gz` on Solaris or `IWOVha.5.5.2.BuildXXXX.tar.gz` on AIX—from the distribution media into `iw-home`.

4. Uncompress the installation file:

On Solaris:

```
% gunzip -c IWOVha-sol.5.5.2.BuildXXXX.tar.gz | (tar xvpf -)
```

On AIX:

```
% gunzip -c IWOVha-aix.5.5.2.BuildXXXX.tar.gz | (tar xvpf -)
```

This creates several HA-specific subdirectories in `iw-home`.

5. Go to `iw-home/ha/install` and run the `iwinstallha` command:

```
% iwinstallha
```

6. Restart `iwserver`:

```
% /etc/init.d/iw.server start
```

## Configuring TeamSite HA Watchdog

Configuring TeamSite HA requires that you edit the `iw.powerfail` and `iw.processfail` scripts to execute tasks that are relevant and specific to your installation. Details are as follows.

### **iw.powerfail**

The default `iw.powerfail` script shown here is shipped with TeamSite. In its current form, it only logs its own name (`iw.powerfail`) when executed. It also contains a commented example of how you could configure the script to run the `iwsi` program, and send email to a system administrator when the script executes. You can configure this script to perform any action upon execution; the only requirement is that you use Perl syntax compatible with Perl Release 5.00503. All results returned by `iw.powerfail` are logged in `iwserver.log`.

To force `iwtock` to exit rather than start the TeamSite server after `iw.powerfail` executes, specify an `iw.powerfail` exit value of 127. This feature is included for scenarios in which TeamSite should not restart automatically following a power failure.

```
use File::Basename;
print(basename($0) . "\n");

#
Use this script to execute processes that can clean up after a powerfail
crash.
#
This script is executed when the Watchdog daemon determines that the
system was not taken down cleanly, and the daemon is itself beginning
execution.
Some of the things that might be tried:
#

iwsi

#
You may also want to mail your system administrator at this point:
#

#use Mail::Send;
#$msg = new Mail::Send Subject=>'TeamSite problem', To=>'admin,root';
#$mfh = $msg->open;
#print $mfh "Please address TeamSite issues at your earliest convenience";
#$mfh->close;

#
If, after executing the backing store utilities, you do not wish to
continue bringing up the system, then exit this script with a 127.
127 indicates to the daemon that it is not to continue with the bringup.
#

#exit 127
```

### **iw.processfail**

The default `iw.processfail` script shown here is shipped with TeamSite. In its current form, it only logs its own name (`iw.processfail`) when executed. It also contains a commented example of how you could configure the script to run the `iwsi` program, and send email to a system administrator when the script executes. You can configure this script to perform any action upon execution; the only requirement is that you use Perl syntax compatible with Perl Release 5.00503. All results returned by `iw.processfail` are logged in `iwserver.log`.

To force `iwtock` to exit rather than restart the TeamSite server after `iw.processfail` executes, specify an `iw.processfail` exit value of 127. This feature is included for scenarios in which TeamSite should not restart automatically following a process failure.

```
use File::Basename;
print(basename($0) . "\n");
#
Use this script to execute processes that can clean up after a TeamSite
crash after the system has begun processing data.
#
This script is executed when the Watchdog daemon determines that the
system was not taken down cleanly, and the daemon has already begun
observing the execution of TeamSite. Some of the things that might be
tried:
#
iwsi
#
You may also want to mail your system administrator at this point:
#
#use Mail::Send;
#$msg = new Mail::Send Subject=>'TeamSite problem', To=>'admin,root';
#$mfh = $msg->open;
#print $mfh "Please address TeamSite issues at your earliest convenience";
#$mfh->close;
#
If, after executing the backing store utilities, you do not wish to
restart the system, then exit this script with a 127. 127 indicates
to the daemon that it is not to restart the server. The server will
not restart under NT at all.
#
#exit 127
```

## Starting and Stopping the Server Under HA Watchdog

You can manually start and stop `iwserver` under TeamSite HA Watchdog just as you would under the base version of TeamSite. See Chapter 7, “Managing the TeamSite Server” for more information. On any list of active system processes, `iwtock` appears as `iwperl` (you will not see a list entry called `iwtock`).

## Uninstalling TeamSite HA Watchdog

The following sections describe how to uninstall TeamSite HA and revert to the base version of TeamSite.

**Caution:** The following procedure deletes the entire ha subdirectory. If you plan to restart TeamSite HA later, you should back up the contents of ha to a different location before you start this procedure.

1. Log in as **root**.

2. Stop **iwserver**:

```
% /etc/init.d/iwserver stop
```

3. Go to **iw-home/ha/install** and run the **iwuninstallha** command:

```
% iwuninstallha
```

You will be prompted to confirm that you want to uninstall TeamSite HA. Answer **yes** to continue uninstalling.

4. Restart **iwserver** after **iwuninstallha** finishes executing:

```
% /etc/init.d/iw.server start
```

## Related Documentation

See the **iws1** documentation in *TeamSite Command-Line Tools* for more information about TeamSite HA Watchdog.

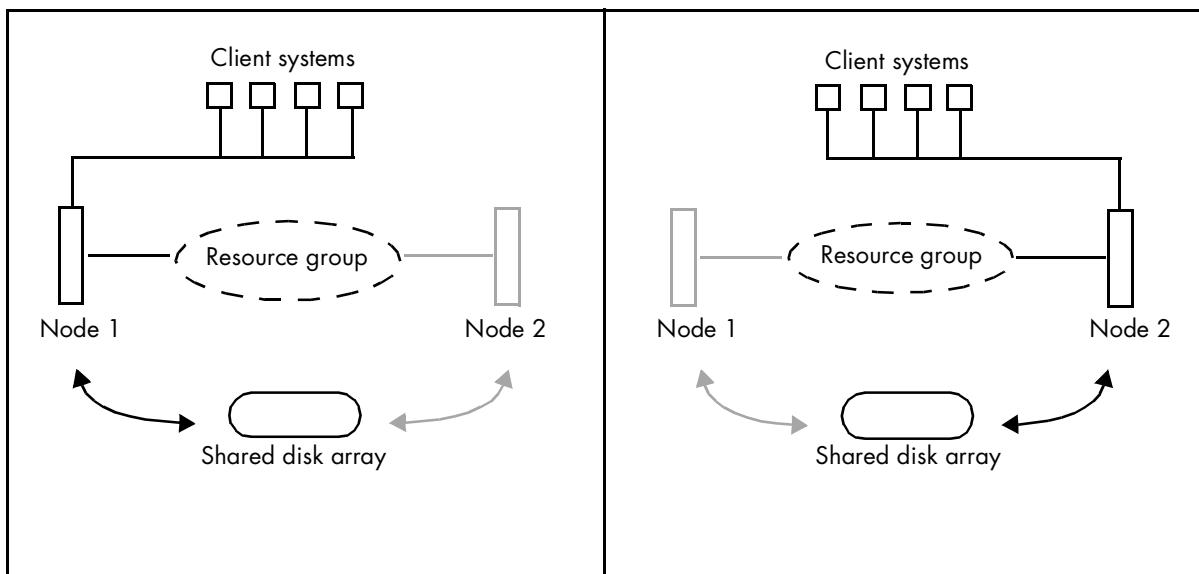
## HA Hot Standby

### About HA Hot Standby

High Availability Hot Standby integrates TeamSite with Sun Cluster 2.2 to provide uninterrupted access to the TeamSite server. HA Hot Standby is not available on AIX.

High Availability Hot Standby uses a two system (or *node*), cluster on Sun Solaris. In this environment, a single *resource group* is created that contains resources needed for TeamSite operation (IP address, location of backing store and backing store copy, network name, TeamSite services). While each node can own and manage cluster resources and a shared disk array containing the TeamSite backing store and a copy of the backing store, only one node can access these at a time. In other words, a node is always active while the other waits passively for a failover to occur.

When any TeamSite service on the active node fails, the passive one becomes active, taking ownership of the cluster's resources and using the last known good copy of the backing store as its backing store. Client requests are then redirected to the currently active node until the failed system is brought back online.



**Normal Operation:** Node 1, the primary node, owns the resource group containing such resources as a shared IP address, location of the backing store and backing store copy, network name, and TeamSite. The backing store and backing store copy are located on a shared disk array connected to both nodes via SCSI.

**Failover:** A failure occurs on the primary node. Node 2 takes ownership of the resource group and uses the copy of the backing store as its backing store. Client requests are routed to the second node until the failed node is brought back online.

## Installing TeamSite and High Availability Hot Standby

### Before You Begin

Before you install, read through the following guidelines for installing TeamSite and HA hot standby. Note that TeamSite integration with Sun Cluster is supported for 2-node cluster setup only.

- Install the Sun Cluster 2.2 software and a volume manager (for example, Veritas Volume Manager). Include the basic cluster inter-connectivity, and verify failover capability.
- Create disk sets, disk groups, and logical hosts. Refer to the Sun Cluster documentation for information about creating logical hosts, and to the Volume Manager documentation for information about creating disk sets and disk groups.
- Create a logical host that masters multiple disk sets.
- Be aware that you can configure TeamSite data service to run on only one logical host.
- When you install TeamSite (instructions below), install it on a local disk with the same directory path in both cluster nodes. For example, if TeamSite is installed in `/usr/iw-home` on Node 1, TeamSite must also be installed in `/usr/iw-home` on Node 2.

Installing on the local disk is important for enabling rolling updates. However, the system administrator should keep some data stored on the local disk in sync between the two nodes, including `iw.cfg`, `available_templates.cfg`, and all `.wft` files.

- Install the Sun Cluster software in `/opt/SUNWcluster` on both nodes.
- Append the root user PATH environment variable with `/opt/SUNWcluster/bin` in `/etc/profile` or `/etc/cshrc`.
- Familiarize yourself with basic Sun Cluster administration tasks and have access to the product documentation.

### Installing TeamSite in the Sun Cluster Environment

To install TeamSite in the Sun Cluster environment:

1. Stop the Sun Cluster software as described in the Sun Cluster Administration documentation.
2. Start the installation procedure described in Chapter 2, “Installing TeamSite.”
3. When the installation program prompts you for the backing store location, enter a path that points to a local disk that is *not* on the shared disk array.

4. Complete the installation and update the license information in `iw.cfg`.

5. Remove or rename the following files:

`/etc/rc0.d/K16iw.server`  
`/etc/rc0.d/K99iw.local`  
`/etc/rc3.d/K16iw.server`  
`/etc/rc3.d/K99iw.local`  
`/etc/rc3.d/S99iw.local`  
`/etc/rc3.d/S16iw.server`

Removing these files prevents the `init` process from starting TeamSite as part of the system startup.

The two previous steps are necessary because starting and stopping TeamSite in a clustered environment is controlled by the Cluster Software depending on which node masters the logical host on which the TeamSite Data Service is running.

6. If this is the first time TeamSite has been installed on this node, reboot the machine.

7. Start TeamSite by running

**% /etc/init.d/iw.server start**

and verify that the server starts successfully.

8. Stop TeamSite by running

**% /etc/init.d/iw.server stop**

This completes the TeamSite installation on Node 1.

9. Repeat steps 2 through 9 to install TeamSite on Node 2.

10. Start the Cluster software on both nodes by executing:

**% scadmin startcluster local\_node\_name cluster\_name**

on the first node, and:

**% scadmin startnode**

on the second node. Refer to the Sun Cluster documentation for detailed information about starting the Cluster software.

11. Execute the `df -k` command to verify that logical hosts are running and that all disk sets are mounted correctly.

This completes the installation and configuration of the TeamSite software on Nodes 1 and 2. Next, you will need to apply the TeamSite-Sun Cluster patch, as described below.

## Applying the TeamSite-Sun Cluster Integration Patch

To apply the TeamSite-Sun Cluster integration patch:

1. Log in as root on Node 1.
2. Set the read and write permissions for root on this directory.
3. Copy the patch to `iw-home/cluster`. (TeamSite-Sun Cluster Integration is supplied as a patch containing a single tar file.)
4. Run:  
**% chmod +x iwov\***
5. Log in as root on the node that currently masters the logical host on which TeamSite will be initially configured to run.
6. Create an `iw-store` directory on the disk set that you will use as the primary backing store. If you are migrating TeamSite from a non-clustered environment to a clustered environment, copy the existing backing store contents to the newly created directory.
7. Open the `/etc/defaultiwstore` file and modify the existing backing store path to the newly created backing store path created in step 11.
8. Ensure `/opt/SUNWcluster/bin` has been appended to the PATH environment variable.
9. Navigate to the `iw-home/cluster` directory.
10. Run:

**% ./iwovreg**

and enter the logical host name, the names of other data services (separated by commas) on which you want `hainterwoven` to depend, and press Y (yes) to create the `hainterwoven` data service.

11.Run **hareg** to verify that the **hainterwoven** data service is created successfully (the output shows **hainterwoven** in the “off” state).

12.Run:

```
% hareg -y hainterwoven
```

to change the **hainterwoven** data service to the “on” state.

13.Run **df -k** to verify that the TeamSite file system is mounted.

14.Log in as root to the standby node (the node to which TeamSite would failover if a failure occurred on the other node on which it is currently running).

15.Open the **/etc/defaultiwstore** file and modify the backing store path to point to the directory on the shared disk array.

16.If you want TeamSite to use a copy of the backing store after a failover occurs, modify the **/etc/defaultiwstore** file to point to a directory that is on a disk set other than the primary backing store.

17.In either case, the modified path should be on a disk set that is mastered by the logical host specified when registering **hainterwoven** service.

## Verifying the **hainterwoven** Data Service Failover

After waiting about 10 minutes for the fault monitor to begin probing, you can verify the **hainterwoven** Data Service Failover by either of the following methods:

- Execute:

```
% /etc/init.d/iw.server stop
```

to stop the TeamSite server on the node on which it currently runs. Wait for a couple of minutes and verify that TeamSite server is started on the second node.

- Run:

```
% iwfreeze +60
```

then run the Sun Cluster administrator utility

```
% scadmin switch cluster_name destination_host logical_host
```

to switch the logical host on which the `hainterwoven` data service is running from the current physical host to the other physical host.

## Additional Information

- The fault monitor script `iwov_fault_monitor` checks the TeamSite server (`hainterwoven` data service) every 30 seconds (this value can be modified). The fault monitor script waits for 120 seconds for TeamSite server to initialize when the `hainterwoven` service is started.
- Method scripts `iwov_start` and `iwov_start_net` are configured to run for a maximum time out value of 60 seconds (that is, TeamSite is expected to start within 60 seconds).
- Similarly, method scripts `iwov_stop` and `iwov_stop_net` are configured to run with a maximum timeout value of 120 seconds (that is, TeamSite is expected to stop within 120 seconds). However, depending on the activity and the backing store size, stopping the TeamSite server can take significantly longer than 120 seconds.

If you want to change this timeout value for these method scripts, contact Interwoven Professional Services for assistance (<http://www.interwoven.com/services/>).

- Client machines that mount the TeamSite file system should use the logical host name in which `hainterwoven` data service is configured to run, rather than individual cluster node names. For example, if the `hainterwoven` data service is configured to run on a logical host called `loiwov`, the entry in the client machine's `/etc/vfstab` should be:

```
loiwov:/iwserv - /mount_point nfs - yes vers=2,bg
```

where `mount_point` is the location on which the file system is to be mounted.

- TeamSite requires a webserver to use the GUI. Scripts for managing the `hainterwoven` data service do not manage or control the availability of that webserver.
- If you plan to switch the `hainterwoven` service from one node to another, use the `iwfreeze` command before running the `haswitch` or `scadmin switch` commands.
- After configuring TeamSite as a data service in your cluster environment, it is recommended that you control it using the Sun Cluster tool `hareg`. For example, start the service with the

```
% hareg -y hainterwoven
```

command and stop it with

```
% hareg -n hainterwoven
```

- If the Sun Cluster software is installed in a directory other than /opt/SUNWcluster, all the scripts installed under `iw-home/cluster` directory need to be modified before creating and using the `hainterwoven` data service.

For more information, contact Interwoven Client Services  
(<http://www.interwoven.com/services/>).

## Appendix D

# Internationalization

---

TeamSite is engineered with your global enterprise in mind. This includes internationalizing the TeamSite server to support multibyte languages and locales at the operating system, and localizing the WebDesk user interface and documentation. Internationalized TeamSite supports the following needs:

- International user data—Enables users to enter data, content, and field values in English, Traditional Chinese, Simplified Chinese, French, German, and Japanese.
- Localized operating system—The TeamSite server runs on any one of the following localized operating systems: English, French, German, and Japanese (one locale per instance of `iwserver`).
- Localized user interface—The WebDesk GUI has been localized in French, German and Japanese.
- Localized file names—You are no longer restricted to having file and directory names in ASCII character encoding. File, directory, branch, workarea, and edition names can have Japanese names on Japanese servers, German names on German servers, and French names on French servers.
- Continued support for processing of non-English metadata and Templating content (introduced in TeamSite 4.2.1 and 4.5.1).

## Supported Client and Server Platforms

The client connecting to the TeamSite server must use the same language as the server (they can be different locales of the same language). For example, running WebDesk on French Windows 98 connected to a Solaris 2.7 TeamSite server running in the French Latin 1 locale (`fr`) is supported. However, if that same French Windows 98 client logged into a Windows 2000 Japanese TeamSite server, and added files with names containing French characters, those files would not be supported by the TeamSite server due to limitations with the native operating system and handling of characters outside of its code pages.

Team Site 5.5.2L supports client/server interaction by clients and servers running in the following configurations:

## Servers

- Solaris 2.7 running in the following locales:
  - English US-ASCII (C)
  - French Latin 1 (fr)
  - German Latin 1 (de)
  - Japanese Shift-JIS (ja\_JP.PCK)
  - Japanese EUC-JP (ja)
- Solaris 2.6 and 2.8 running in the US-ASCII locale (C)
- AIX 5.1 running in the following locales:
  - English US-ASCII (C)
  - French Latin 1 (fr\_FR)
  - German Latin 1 (de\_DE)
  - Japanese Shift-JIS (Ja\_JP.IBM-932)
  - Japanese EUC-JP (ja\_JP)

## Clients

- Windows 98 (US English, French, German, Japanese)
- Windows NT (US English, French, German, Japanese)
- Windows 2000 (US English, French, German, Japanese)
- Solaris 2.6, 2.7, and 2.8 US (English)
- MacOS 9.x (US English, Japanese)

## Browsers

- Internet Explorer 5.x (Internet Explorer 6 is not supported)
- Netscape 4.76 through 5.x (Netscape 6 is not supported, Netscape is not supported on MacOS 9.x)

Refer to the table on page 27 for more information about browser compatibility.

**Note:** Appendix E, “Client/Server Compatability” contains a series of tables that contain information about client/server/language compatibility for all supported platforms; WebDesk UI, Launchpad, and Templating UI; file, directory, backing store, branch, and workarea names; and Templating content and metadata.

## Supported TeamSite Server Locales

The following table describe the supported TeamSite server locales:

Language	Server Locale Supported
Japanese	Solaris: ja_JP.PCK and ja (also known as ja_JP.EUC-JP) AIX: ja_JP and Ja_JP.IBM-932
German	Solaris: de AIX: de_DE
French	Solaris: fr AIX: fr_FR
English	C (C locale)

The TeamSite `iw.cfg` file now contains a `server_locale` entry in the `[iwserv]` section. The entry specifies the locale in which current execution of the TeamSite server (`iwserv`) is running. For detailed information about the `server_locale` setting, refer to “Configuring the TeamSite Server Locale” on page 163.

## Supported Content

TeamSite supports non-ASCII characters in branch, area, directory, `vpath`, and file names in addition to the contents of a file.

## Localization Overview

The following sections list whether or not major TeamSite features have been translated. These features are described throughout the TeamSite documentation.

## What's Been Translated?

The following TeamSite 5.5.2L features have been translated into French, German, and Japanese:

- TeamSite login screen
- WebDesk user interface and online help
- WebDesk submit workflow
- VisualFormat
- LaunchPad applet
- Java Templating client installer
- TeamSite Templating (browser-based and Java Templating) and Visual Format
- SmartContext Editing (SCE)
- The following printed documentation (including PDF versions on TeamSite CD-ROM):
  - *TeamSite Author's Guide*
  - *TeamSite User's Guide*
  - *TeamSite Templating User's Guide*

## What's Not Been Translated?

The following TeamSite 5.5.2L features are available in English only:

- TeamSite installer
- TeamSite Templating server installer
- WebDesk Pro user interface and online help (see note below)
- LaunchPad application (no longer supported on any operating system)
- Metadata capture form
- Command-line Tool user interface
- The remainder of the printed documentation and Release Notes
- WorkflowBuilder application (WorkflowBuilder only runs on US clients and servers)

WebDesk Pro GUI elements, including buttons and drop-down menus, retain English names but may look slightly different because all HTML pages of our browser-based GUI are UTF-8 encoded, even for US English installations. Your client browsers may therefore choose different fonts to render UTF-8 encoded HTML pages.

TeamSite users can interact with the GUI using any one of the supported languages, but the TeamSite server must be listed in “Supported TeamSite Server Locales” on page 317.

## Limitations and Assumptions

- An internationalized TeamSite server does not mean that your TeamSite server can be run in multiple locales concurrently. The TeamSite server can run in any supported locale, but one locale at a time.
- It is expected that the locale in which the TeamSite server runs is the same locale as the rest of file system and server operating systems. Consider the following scenario:
  - a. You have a file server which runs in `ja` (Japanese Extended UNIX Code) locale, with a hierarchy of file and directory structures with names encoded in Japanese EUC.
  - b. You install and run your TeamSite server on this file server.
  - c. You use the file system interface to migrate your existing hierarchy of files and directories into TeamSite’s Intelligent File System (`/iwmnt`).
  - d. The TeamSite server must run in a `ja` locale for these file and directory names to be processed correctly. If you change the locale to `ja_JP.PCK` (Japanese Shift-JIS, `Ja_JP.IBM-932` on AIX) before TeamSite server is started, the TeamSite server would interpret the imported file and directory names as `ja_JP.PCK` (or `Ja_JP.IBM-932`) encoded. This is not a supported scenario.
- Mixed-locale file systems are not supported. For example, a scenario where a parent directory has directory names encoded in `ja_JP.PCK` (Japanese Shift-JIS, `Ja_JP.IBM-932` on AIX), and child directories have file names encoded in `ja` (`ja_JP` on AIX) is not supported.
- If TeamSite server is running on a German operating system using a German Latin1 locale, it is possible to create a branch or workarea on the TeamSite Intelligent File System with Japanese names using the TeamSite GUIs. However, when viewed with the file system

interface, these Japanese names would appear as illegible characters because the server is running in a Latin1 locale and does not include the Japanese character set. This is not a supported scenario.

Note that this scenario is supported for Metadata because Metadata entered using the TeamSite GUIs does not interact with server operating system. Any data that is interchanged with the server operating system (including VPATHs) are only meaningful if they are within the server locale's encoding.

- If TeamSite Intelligent File System is functioning as a networked file server, it is expected that all other networked file system clients (for example, NFS clients) are operating in the same locale as the TeamSite Intelligent File System file server.

Currently, NFS does not enforce this restriction and therefore enables NFS clients to be in a different locale than the NFS server. However, NFS protocol does not do encoding conversion. Therefore, file and directory attributes (including names) are passed through in binary format. This would not work for TeamSite IFS functioning as file server because it does encoding conversion from and into UTF-8 based on the server file system's locale.

## Backing Stores and Character Encoding

User-defined backing stores which are named using multibyte characters, must have a corresponding entry in the `iw.cfg` file. Refer to “Creating Multiple Backing Stores” on page 261 for detailed information.

## About UTF-8

UTF-8 is the 8-bit encoding format for Unicode. Unicode is a system for exchanging, processing, and displaying diverse written languages. Unicode supports the principal written languages of the world as well as many classical languages.

## CCI URLs with Multibyte Characters

When constructing URLs for the Casual Contributor Interface when parameters contain multibyte characters, make sure that these parameters are URL- and UTF-8-encoded. Multibyte characters should be URL-encoded based on their Unicode representation in UTF-8.

For example, the URL to edit the file:

```
/archive/main/WORKAREA/area/↗.html
```

would be:

```
/iw/webdesk/edit?vpath=/archive/main/WORKAREA/area/%e5%ba%9c.html
```

since the Japanese character ↗ is Unicode character U+30D5, which is encoded as the bytes 0xe5 0xba 0x9c in the UTF-8 format.

## Interfacing with Localized Operating Systems

The internationalized TeamSite server's virtualized Intelligent File System (IFS) functions the same way a regular file system does on localized operating systems. For example, if TeamSite runs on a server that is running in the EUC-JP locale, the TeamSite IFS is displayed and functions as an EUC-JP encoded file system.

To achieve this, TeamSite system calls to the operating system are converted from UTF-8 encoded textual data (for example, VPATH information) into the locale of `iwserver` (as defined by the `server_locale` setting in `iw.cfg`). In most cases, this is the same as the operating system's native locale. The conversion is also required when operating system information is returned to TeamSite.

If the TeamSite server is run in a different locale than the host operating system's locale, the TeamSite virtual file system would use a different encoding locale compared to the rest of host server's file systems. By default, the TeamSite server locale uses the native locale of the host operating system.

## Accessing the Localized Interface

To display the localized (French, German, or Japanese) WebDesk interface, you must change your browser's language settings to the appropriate language.

To display the localized (French, German, or Japanese) LaunchPad and Java Templating interfaces, your client operating system must be in the same locale as the interface you want to display.

## CLT Internationalization

Command-line tools are now locale-sensitive such that arguments passed into the CLT as textual arguments—including submit comments and VPATH specifications—can be text characters from any of the supported languages (English, French, German, or Japanese). For example, if you typed submit comments into the `iwsubmit` CLT in Japanese, the character encoding of the submit comments would depend on the locale under which `iwsubmit` was executed.

When CLTs are executed, the locale is determined by referencing the `LANG` environment variable. Based on this locale, it determines how to interpret character encoding of the textual arguments passed from the command-line.

## CGI Internationalization

Since the 4.2.1 release, TeamSite CGIs have used UTF-8 encoding to serve pages to browsers. This causes browsers to return data to TeamSite encoded in UTF-8. This enables TeamSite to support metadata in any language or native encoding. TeamSite 5.5.2L uses this same methodology with VPATHs in addition to metadata.

## Specifying File Encoding of Text Files

All browsers rely on default settings to “guess” the encoding of web pages whose encoding is not explicitly declared. If the browser’s default setting is different than that of the actual encoding of the page passed to the browser, the browser may render the page incorrectly.

Therefore, the best practice is for your web pages to always declare their encoding. This prevents your browser from guessing incorrectly when you use TeamSite, and ensures that your web site viewers' browsers will not have to guess which encoding they should use.

For HTML documents, Smart Context Editing (SCE) honors the encoding specified by the `charset` parameter in either a `Content-Type` HTTP header or in an HTML META tag. For example:

- `Content-Type: text/plain; charset=UTF-8`
- `<META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=UTF-8">`

To display multibyte characters in non-HTML text documents in SCE with the desired character encoding, the content webserver must be configured to return a `Content-Type` HTTP header that specifies the encoding, for example:

`Content-Type: text/plain; charset=UTF-8`

If the `charset` is not specified—either by the content webserver's `Content-type` HTTP header, or by the `charset` tag within the file—SCE assumes that the document is encoded in ISO-8859-1, which may cause the document to be displayed with “garbage” characters.

To solve the issue of text files that do not specify their encoding, TeamSite 5.5.2L has introduced a new configuration file called `file_encoding.cfg`. Please refer to Appendix B, “Specifying Content Encoding” for detailed information about creating/configuring settings in `file_encoding.cfg`.

## Text Editor Encodings

The following table shows the default settings for various text editors.

<b>Text Editor</b>	<b>Platform</b>	<b>Default Encoding</b>	<b>To Save as UTF-8 Encoding:</b>
vi	Solaris	Depends on the locale of vi's active process.	Cannot save or render text as UTF-8.
emacs	Solaris	Depends on the locale of emacs's active process.	Cannot save or render text as UTF-8.

## Behavior of Netscape Navigator

If a Netscape browser finds a UTF-8 page, it uses UTF-8 as its default encoding for pages that do not specify their encoding. This may cause the browser to display pages incorrectly if the user browses pages that do not specify their encoding, or creates pages without specifying the encoding.

### Scenario 1

1. A Japanese user goes to a Japanese site which does not specify its encoding. Netscape defaults to Japanese (Auto-Detect).
2. The Japanese user logs into TeamSite (UTF-8 pages). Netscape switches to UTF-8.
3. The Japanese user opens a new window and returns to the Japanese site which does not specify its encoding. Now Netscape defaults to UTF-8.

This would not happen if the site specified the encoding of its web pages.

### Scenario 2

1. A Japanese user logs into TeamSite (UTF-8 pages). Netscape switches to UTF-8.
2. The Japanese user's content in TeamSite does not include the 'Content-type' META tag.
3. Upon entering SmartContext QA, Netscape tries to render the content as UTF-8, which is probably wrong. The solution to this problem is to always specify the encoding for all HTML content.

## Configuring Netscape for Multibyte Characters

Complete the following procedure if you are using a Netscape browser to display multibyte characters:

1. Open your Netscape browser.
2. Select **Edit > Preferences** to display the Preferences dialog box.
3. Click **Appearance > Fonts** to display the Fonts settings.
4. Set the **For the Encoding** field to Unicode.
5. Set the **Variable Width Font** field to a font which supports the language you want to use.
6. Set the **Fixed Width Font** field to a font which supports the language you want to use.
7. Click the **Use my default fonts, overriding document-specified fonts** option.
8. Click **OK**.

If this procedure does not deliver the expected results (that is, certain characters are not displayed properly), try the following procedure:

1. Select **View > Character Set > Set Default Character Set**.
2. Select **View > Character Set > Unicode (UTF-8)**.

## Usage Scenarios

The following examples illustrate some of the advantages of using TeamSite in a global enterprise. Note that a branch scenario could also apply to a workarea, directory, or file operation (for example, New Branch, New Workarea, and Import File). Scenarios can also be applied to other locales.

**Scenario 1**

1. The TeamSite server is running in the `ja_JP.PCK` (Shift-JIS) locale on Solaris 2.7.
2. You create a branch with a Japanese name using WebDesk Pro running on Japanese Windows NT. This branch is created in the TeamSite Intelligent File System with Shift-JIS encoding.
3. You can navigate this branch with the Japanese name using WebDesk or WebDesk Pro.
4. You can also log on to the server machine and access this branch with Japanese name using the file system interface.

**Scenario 2**

1. The TeamSite server is running in the `ja_JP.PCK` (Shift-JIS) locale on Solaris 2.7.
2. Your TeamSite Administrator copies a directory from the file system into the TeamSite Intelligent File System. This directory contains file and directory names with Japanese encoded names.
3. Your TeamSite Administrator creates a file in the TeamSite Intelligent File System with a Japanese (Shift-JIS)encoded name.
4. WebDesk Pro and WebDesk users (on any client platform) can view and access this directory (and corresponding files) with a Japanese name.

**Scenario 3**

1. You install and run TeamSite on a Japanese Solaris system in the `ja_JP.PCK` locale. The file server for this system operates in the Shift-JIS locale (`ja_JP.PCK` locale on Solaris 2.7 is a Shift-JIS locale).
2. You create a branch with a Japanese name using the TeamSite GUI.

This branch is displayed in `/iwmnt` as a directory with a Shift-JIS encoded directory name and is displayed in all typical file system operations with a Shift-JIS encoded directory name just like the other files and directories in the file system.

### **Scenario 4**

You install and run TeamSite on a Japanese Solaris system in a `ja_JP.PCK` locale. The file server for this system operates in the `Shift-JIS` locale.

3. You copy an existing hierarchy of files and directory structures into a workarea called `isuzuki` within `/iwmnt`.
4. You use one of the TeamSite GUIs to access the `isuzuki` workarea.

The file and directory hierarchy is displayed with Japanese directory and file names, and is correctly referenced in the TeamSite GUI.

### **Scenario 5**

1. You install and run TeamSite on a Japanese Solaris system in a `ja_JP.PCK` locale.
2. Using one of the TeamSite GUIs, you submit a file and comments in Japanese.
3. You use the `iwsubmit` CLT to view the extended attributes of the file.

The Japanese submit comments are displayed correctly on the command-line. After executing the submit, the same submit comments are displayed correctly in history log of the file submitted.



## Appendix E

# Client/Server Compatibility

---

	Server OS				
	AIX 5.1 C locale	AIX 5.1 de_DE locale (German)	AIX 5.1 fr_FR locale (French)	AIX 5.1 ja_JP (Japanese)	AIX 5.1 Ja_JP.IBM-932 (Japanese)
Client OS					
Windows 98 French			x		
Windows 98 German		x			
Windows 98 Japanese				x	x
Windows 98 US	x				
Windows NT French			x		
Windows NT German		x			
Windows NT Japanese				x	x
Windows NT US	x				
Windows 2000 French			x		
Windows 2000 German		x			
Windows 2000 Japanese				x	x
Windows 2000 US	x				
MacOS 9.0 Japanese				x	x

	<b>Server OS</b>				
	<b>AIX 5.1 C locale</b>	<b>AIX 5.1 de_DE locale (German)</b>	<b>AIX 5.1 fr_FR locale (French)</b>	<b>AIX 5.1 ja_JP (Japanese)</b>	<b>AIX 5.1 Ja_JP.IBM-932 (Japanese)</b>
MacOS 9.0 US	x				
Solaris 2.6, 2.7, 2.8 (C locale)	x				
<b>WebDesk UI, Launchpad, and Templating UI</b>					
French			x		
German		x			
Japanese				x	x
English	x	x	x	x	x
<b>File, directory, store, workarea, and branch names</b>					
French			x		
German		x			
Japanese				x	x
English	x	x	x	x	x
<b>Templating content and metadata</b>					
French	x	x	x	x	x
German	x	x	x	x	x
Japanese	x	x	x	x	x
English	x	x	x	x	x
Traditional Chinese	x	x	x	x	x
Simplified Chinese	x	x	x	x	x

	Server OS			
	Windows 2000 US	Windows 2000 Japanese	Windows 2000 French	Windows 2000 German
Client OS				
Windows 98 French			x	
Windows 98 German				x
Windows 98 Japanese		x		
Windows 98 US	x			
Windows NT French			x	
Windows NT German				x
Windows NT Japanese		x		
Windows NT US	x			
Windows 2000 French			x	
Windows 2000 German				x
Windows 2000 Japanese		x		
Windows 2000 US	x			
MacOS 9.0 Japanese		x		
MacOS 9.0 US	x			
Solaris 2.6, 2.7, 2.8 (C locale)	x			

WebDesk UI, Launchpad, and Templating UI				
French			x	
German				x
Japanese		x		

	<b>Server OS</b>			
	<b>Windows 2000 US</b>	<b>Windows 2000 Japanese</b>	<b>Windows 2000 French</b>	<b>Windows 2000 German</b>
English	x	x	x	x
<b>File, directory, store, workarea, and branch names</b>				
French			x	
German				x
Japanese		x		
English	x	x	x	x
<b>Templating content and metadata</b>				
French	x	x	x	x
German	x	x	x	x
Japanese	x	x	x	x
English	x	x	x	x
Traditional Chinese	x	x	x	x
Simplified Chinese	x	x	x	x

	<b>Server OS</b>				
	<b>Solaris 2.6, 2.7, 2.8 C locale</b>	<b>Solaris 2.7 de locale (German)</b>	<b>Solaris 2.7 fr locale (French)</b>	<b>Solaris 2.7 Shift-JIS locale (Japanese)</b>	<b>Solaris 2.7 EUC-JP locale (Japanese)</b>
<b>Client OS</b>					
Windows 98 French			x		

	Server OS				
	Solaris 2.6, 2.7, 2.8 C locale	Solaris 2.7 de locale (German)	Solaris 2.7 fr locale (French)	Solaris 2.7 Shift-JIS locale (Japanese)	Solaris 2.7 EUC-JP locale (Japanese)
Windows 98 German		x			
Windows 98 Japanese				x	x
Windows 98 US	x				
Windows NT French			x		
Windows NT German		x			
Windows NT Japanese				x	x
Windows NT US	x				
Windows 2000 French			x		
Windows 2000 German		x			
Windows 2000 Japanese				x	x
Windows 2000 US	x				
MacOS 9.0 Japanese				x	x
MacOS 9.0 US	x				
Solaris 2.6, 2.7, 2.8 (C locale)	x				

WebDesk UI, Launchpad, and Templating UI					
French			x		

	<b>Server OS</b>				
	<b>Solaris 2.6, 2.7, 2.8 C locale</b>	<b>Solaris 2.7 de locale (German)</b>	<b>Solaris 2.7 fr locale (French)</b>	<b>Solaris 2.7 Shift-JIS locale (Japanese)</b>	<b>Solaris 2.7 EUC-JP locale (Japanese)</b>
German		x			
Japanese				x	x
English	x	x	x	x	x
<b>File, directory, store, workarea, and branch names</b>					
French			x		
German		x			
Japanese				x	x
English	x	x	x	x	x
<b>Templating content and metadata</b>					
French	x	x	x	x	x
German	x	x	x	x	x
Japanese	x	x	x	x	x
English	x	x	x	x	x
Traditional Chinese	x	x	x	x	x
Simplified Chinese	x	x	x	x	x

	<b>Server OS</b>	
	<b>Windows NT US</b>	<b>Windows NT Japanese</b>
<b>Client OS</b>		
Windows 98 French		
Windows 98 German		

	<b>Server OS</b>	
	<b>Windows NT US</b>	<b>Windows NT Japanese</b>
Windows 98 Japanese		<b>x</b>
Windows 98 US	<b>x</b>	
Windows NT French		
Windows NT German		
Windows NT Japanese		<b>x</b>
Windows NT US	<b>x</b>	
Windows 2000 French		
Windows 2000 German		
Windows 2000 Japanese		<b>x</b>
Windows 2000 US	<b>x</b>	
MacOS 9.0 Japanese		<b>x</b>
MacOS 9.0 US	<b>x</b>	
Solaris 2.6, 2.7, 2.8 (C locale)	<b>x</b>	
<b>WebDesk UI, Launchpad, and Templating UI</b>		
French		
German		
Japanese		<b>x</b>
English	<b>x</b>	<b>x</b>

<b>File, directory, store, workarea, and branch names</b>		
French		
German		

	<b>Server OS</b>	
	<b>Windows NT US</b>	<b>Windows NT Japanese</b>
Japanese		<b>x</b>
English	<b>x</b>	<b>x</b>
<b>Templating content and metadata</b>		
French	<b>x</b>	<b>x</b>
German	<b>x</b>	<b>x</b>
Japanese	<b>x</b>	<b>x</b>
English	<b>x</b>	<b>x</b>
Traditional Chinese	<b>x</b>	<b>x</b>
Simplified Chinese	<b>x</b>	<b>x</b>

## **Appendix F**

# **Integrating with SiteMinder**

---

This appendix describes the procedure for configuring TeamSite to integrate with the Netegrity SiteMinder Policy Server. The integration enables:

- The TeamSite server to act as another web server in a portal environment controlled by SiteMinder.
- A single sign-on where TeamSite users log in once against SiteMinder and are authorized to access all of its resources (TeamSite and the other web servers in the portal) without having to log in to TeamSite explicitly.

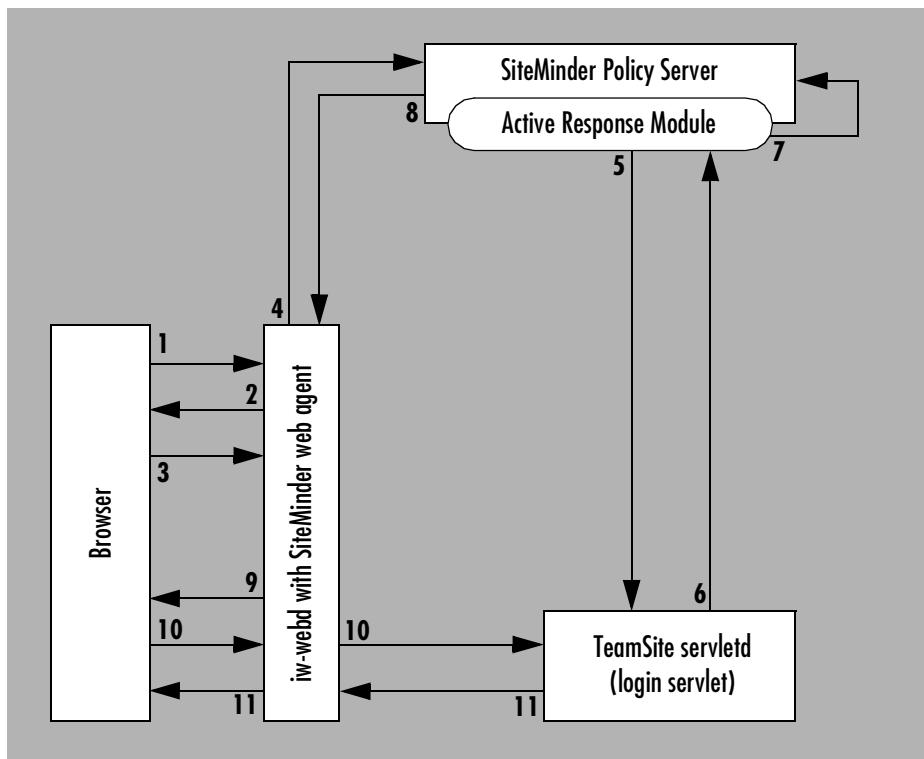
TeamSite for AIX does not support the SiteMinder Policy Server.

## **Requirements**

- TeamSite 5.0.2 or above installed and configured.
- Netegrity SiteMinder 4.5.1 SP2 or above installed and configured.
- Optionally, Netegrity SiteMinder Secure Reverse Proxy (NSSRP) server 1.0 or above installed and configured (see page 343).
- TeamSite and SiteMinder must share the same user database.
- Some familiarity with TeamSite and Netegrity SiteMinder Policy Server and Web Agent products.

## Authentication Overview

After completing the integration described in this appendix, SiteMinder controls access to TeamSite by authenticating the user against its user database and placing an Interwoven IWAUTH cookie in the browser. The typical sequence of data and operation flow is depicted in the graphic and described in the legend that follows:



1. A TeamSite user accesses the TeamSite server as usual (by entering *hostname/iw/* in his or her browser).
2. The web agent intercepts the request and prompts for the user for a user name and password.
3. The user enters his or her user name and password.

4. The web agent passes the login information to the SiteMinder Policy Server which authenticates the user, calls the cookie-placing active response function in the `iwtssmar.so` file.
5. The Active Response module calls the `TeamSite_hostname/iw/webdesk/login` servlet URL with the user name and password as arguments.  
The TeamSite login servlet authenticates the user with the Team Site server and generates the IWAUTH session string.
6. The TeamSite login servlet returns the corresponding Tomcat JSESSIONID cookie.
7. The Active Response module returns the JSESSIONID.
8. The SiteMinder Policy Server sends it to the web agent.
9. The web agent passes the JSESSIONID and a SiteMinder session cookie (SMSSESSION) to the user's browser.
10. The browser passes the JSESSIONID to the servlet engine which passes it to the corresponding login servlet which holds the IWAUTH session string.
11. The TeamSite login servlet places the IWAUTH persistent cookie in the browser and logs the user in using the highest Team Site role associated with the user.

## Integration Procedure Overview

The integration procedure contains four major steps:

- Creating a file on your SiteMinder Policy Server that contains the URL of your TeamSite server.
- Copying the Active Response module (`iwtssmar.so`) to your SiteMinder Policy Server.
- Configuring SiteMinder to recognize TeamSite users.
- Configuring the Reverse Proxy.

The detailed procedures associated with these steps are described in the sections that follow.

## Creating a TeamSite Server URL File

Complete the following procedure to create a text file on your SiteMinder Policy Server that contains the URL of your TeamSite server. You will later create an environmental variable in the `smuser.profile` file which will reference this text file.

1. Change to your SiteMinder home directory, for example:

```
%cd var/siteminder/
```

2. Create a text file that will contain the TeamSite server URL, for example:

```
%vi ts_serverurl.txt
```

3. Enter the URL of the TeamSite server that you want to integrate with SiteMinder, for example:

```
TeamSiteServerURL=http://host_name/iw/webdesk/login
```

Where `host_name` is the hostname of the system on which TeamSite is installed and `domain_name` is the domain of the host. For example, if TeamSite is installed on a system named `factotum`, the entry in the `ts_serverurl.txt` file would be:

```
TeamSiteServerURL=http://factotum.interwoven.com/iw/webdesk/login
```

**Note:** Typically, the TeamSite web server listens at port 80. If a different port is used, you can qualify the URL with `http://host_name:port/iw/webdesk/login` (the `http://` is required).

4. Save and close the file.

## Copying the iwtssmar.so File

The TeamSite 5.5.1 installation program automatically places the Active Response module needed to integrate TeamSite and SiteMinder in the `iw-home/lib` directory on your TeamSite server. The module is in a file is called `iwtssmar.so` and must be copied to your SiteMinder Policy Server.

1. Stop all SiteMinder Policy Server daemons.

Refer to the *SiteMinder Operations Guide* for detailed instructions.

2. Copy the `iwtssmar.so` file from `iw-home/lib` to where the `policyserver_home/bin/smservauth` executable can find it.

For example:

```
%cp iw-home/lib/iwtssmar.so /var/siteminder/bin/
```

### Notes:

- If you are using TeamSite 5.0.2 you must download the file from <http://support.interwoven.com>.
- If the SiteMinder Policy Server is installed on a different computer, use `ftp` to transfer the file to the system where SiteMinder is installed.

## Configuring SiteMinder

Complete the following procedure to create an environmental variable which references the TeamSite Server URL text file created in the previous section, and to configure SiteMinder to recognize TeamSite users. This procedure is performed on the system where the SiteMinder Policy Server is running.

1. Change to the `/usr/home/smuser` directory.
2. Open the `.profile` file in your text editor.
3. Create an environment variable called `INTERWOVEN_TEAMSITE_CONFIG`, for example:  
`export INTERWOVEN_TEAMSITE_CONFIG=/siteminder_home/file_name.txt`

This environment variable must reference the full path of the TeamSite Server URL file. For example, if the name of the file is `ts_serverurl.txt`, and the path to the environment variable is `/var/siteminder/bin/`, the value of the environment variable would be:

```
export INTERWOVEN_TEAMSITE_CONFIG=/var/siteminder/ts_serverurl.txt
```

4. Save and close the `.profile` file.
5. Open the SiteMinder Policy Server user interface where you will create Realms and Rules to protect TeamSite resources.
6. Create the following Netegrity Realms:

<b>Realm</b>	<b>Realm Type</b>
<code>/iw</code>	Protected
<code>/iw/webdesk/login</code>	Unprotected
<code>/iw/index.html</code>	Unprotected

7. Under the `/iw` Realm, create the following Rules:

<b>Rule</b>	<b>Description</b>
<code>Authenticate_iw - iw*</code>	Configured for authenticating event actions.
<code>Protect_iw - iw*</code>	Configured for GET, PUT, POST actions.

The other Realms do not require any rules.

8. Create a Netegrity Response with the name `IWCookieResponse` and the following details:

Field	Entry
Attribute of the Response	<b>WebAgent-HTTP-Cookie-Variable</b>
Attribute Kind	<b>Active Response</b>
Cookie Name	<b>JSESSIONID</b>
Library Name	<b><i>full_path/iwtssmar.so</i></b>
Function Name	<b>activeResponse</b>

9. Create a Netegrity Policy with the following label:

`Add TeamSite users`

10. In the Rules tab, add the following Rules:

- `Authenticate_iw`
- `Protect_iw`

11. For the `Authenticate_iw` Rule, set the Response to `IWCookieResponse`.

## Configuring the Reverse Proxy

When integrating TeamSite and SiteMinder, there are two reverse proxy methods you can use to protect URLs from being accessed:

- Using the Netegrity-provided Apache reverse proxy web agent plug-in (Apache load module) and modifying the `httpd.conf.template` file that is installed by TeamSite. This is the preferred method because it minimizes communication to and from the proxy server and `iw-webd`. This method is described in the next section.
- Using the Netegrity SiteMinder Secure Reverse Proxy (NSSRP). This method is described in “Integrating Using NSSRP” on page 344.

Complete the procedure that corresponds with the method you want to use.

## Integrating Using the Apache Reverse Proxy Web Agent

TeamSite uses the Apache web server as a reverse proxy server, that is, it routes inbound web-based requests internally to other web servers. To achieve a reverse proxy when SiteMinder is used with TeamSite, `iw-webd` must be configured as follows:

- The Apache reverse proxy server must be installed in `/usr/iw-home/iw-webd`.
- Because TeamSite renames the Apache server executable `httpd` to `iwwebd`, you must create a symbolic link from `httpd` to `iwwebd` in the `/usr/iw-home/iw-webd/bin` directory before installing the SiteMinder WebAgent.

After the SiteMinder WebAgent is installed, complete the following procedure to modify the `httpd.conf.template` file on the system where your TeamSite server is installed.

1. Login as `root`.
2. Change to the `iw-home/iw-webd/conf` directory.
3. Open the `httpd.conf.template` file.
4. Add the `LoadModule`, `AddModule`, and `SMInitFile` entries as described in the *SiteMinder Installation Guide*.
5. Modify the `MaxRequestsPerChild`, `MinSpareServers`, `MaxSpareServers`, and `StartServer` entries to match the load in your site.
6. Save and close the `httpd.conf.template` file.
7. Tune the kernel parameters in the `/etc/system` file as described in the “Environment Modifications for Apache” section of the *SiteMinder Installation Guide*.

## Integrating Using NSSRP

Integrating TeamSite and SiteMinder requires that NSSRP is installed and the `proxyrules.xml` file is configured to protect URL access. NSSRP can be installed on the same system where TeamSite is installed, or on a second system.

- If TeamSite and SiteMinder are installed on different systems, begin the NSSRP configuration procedure as described in “Configuring `proxyrules.xml`” on page 346.

- If TeamSite and SiteMinder are installed on the same system, begin the NSSRP configuration procedure as described in the next section.

## Configuring TeamSite Port Numbers

TeamSite uses the Apache web server (`iwwebd`) as a reverse proxy server to route inbound web-based requests to other web servers, the Tomcat application server, and `iwproxy`. TeamSite is normally configured so that `iwwebd` listens at port 80 and Tomcat listens at port 8080 (Tomcat also uses ports 8007 and 8009 for other tasks).

Netegrity uses the same combination of Apache and Tomcat web servers in their NSSRP product resulting in port conflict. To resolve this conflict, configure TeamSite to use different port numbers by completing the following procedure:

1. Change to the `iw-home/etc` directory.

2. Open the `iw.cfg` file.

3. Locate the `teamsite_servlet_ui` section:

```
[teamsite_servlet_ui]
servlet_host=localhost
servlet_port=8080
```

4. Edit the `servlet_port` number to use another port, for example:

```
servlet_port=8980
```

5. Locate the `iwwebd` section.

```
[iwwebd]
default_protocol=http
http_port=80
https_port=443
host=11.420.8.196
```

6. Edit the port numbers, for example:

```
http_port=90
https_port=543
```

7. Save and close the `iw.cfg` file.

8. Change to the `iw-home/servletd/conf` directory.

9. Edit the following files to use the same port numbers you configured in the `iw.cfg` file or new port numbers as noted:
  - `server.xml`—In the `Connectors` section, edit the `value=8080` entry to match the port number you entered in step 4.
  - `workers.properties`—In the `DEFAULT ajp## WORKER DEFINITION` section, edit the `value=8080` entry to match the port number you entered in step 4.
  - `tomcat.properties`—In the `Set the port Apache JServ listens to` section, edit the `port=8007` entry to, for example, `port=8907`.
  - `tomcat.conf`—In the `Change if you run tomcat on a different host` section, edit the `ApJServDefaultPort 8007` entry to match the port number you entered in the `tomcat.properties` file (8907 in this example).

Continue the NSSRP configuration procedure by editing the `proxyrules.xml` file as described in the next section.

### Configuring `proxyrules.xml`

To complete the integration, you must modify the NSSRP `proxyrules.xml` file to match the file shown below. These modifications route all TeamSite URL requests to the TeamSite server after performing the single sign-on. The `proxyrules.xml` file is located in the `NSSRP_home/proxy-engine/conf` directory. The following file assumes that NSSRP and TeamSite are installed on the same system and NSSRP is used to only protect TeamSite resources. Ensure you substitute your actual values for the following variables in the sample:

- `your_iwserver`—Name of your TeamSite server
- `port_number`—Port number used by your TeamSite server
- `NSSRP_home`—Directory where NSSRP is installed

```
<?xml version="1.0"?>
<?cocoon-process type="xslt"?>
<!DOCTYPE nete:proxyrules SYSTEM "file:///NSSRP_home/proxy-engine/conf/dtd/
proxyrules.dtd">
<!-- Proxy Rules -->
<nete:proxyrules xmlns:nete="http://www.netegrity.com/">
 <nete:xprcond>
 <nete:xpr>
 <nete:rule>(.*)</nete:rule>
 <nete:result>http://your_iwserver:port_number$1</nete:result>
 </nete:xpr>
 <nete:xpr>
 <nete:rule>(.*)</nete:rule>
 <nete:result>http://your_iwserver:port_number$1</nete:result>
 </nete:xpr>
 <nete:xpr-default>
 <nete:forward>http://your_iwserver:port_number$0</nete:forward>
 </nete:xpr-default>
 </nete:xprcond>
</nete:proxyrules>
```

If NSSRP is configured to run on a different system than the one TeamSite is installed on, the “*:port\_number*” should be removed as no port-related changes would have been made. NSSRP should then be restarted for this change to take effect.

If you want to use NSSRP to protect multiple web sites in the domain and route traffic to different servers using different rules, a more elaborate *proxyrules.xml* needs to be created. Contact Netegrity’s consulting services for details about this type of implementation.



## Appendix G

# Root Access to TeamSite

---

The following tables describe the TeamSite files that need to be owned by, executed by, or setuid root:

<b>Setuid root</b>	<b>Comments</b>
iw-home/httpd/iw-bin/iwcgi.cgi	For the Set Home Page feature, needs to write to <code>iw-home/local/iwprofiles</code> , which is owned and writable only by root. Also needs to impersonate users to copy files on their behalf.
iw-home/httpd/iw-bin/iw_cgi_wrapper.cgi iw-home/httpd/iw-bin/nph-iw_cgi_wrapper.cgi	Needs to be able to do setuid/setgid to impersonate users while running downstream CGI.
iw-home/httpd/iw-bin/iwmerge.cgi	Needs to impersonate users to write files on their behalf.
iw-home/httpd/iw-bin/iwfileupdownload.cgi	Needs to impersonate users to write files on their behalf.
iw-home/bin/iwgetldapusers	This CLT is used within TeamSite Workflow to get lists of users and roles when LDAP or ActiveDirectory is used instead of TeamSite .uid files.
iw-home/httpd/iw-bin/iwfscckcgi.cgi	Needs to access the backing store directory and files underneath which are accessible only by user root. Only root can run this program, and it requires a password.

<b>Executed by root</b>	<b>Comments</b>
<code>iw-home/bin/iwserver</code>	Needs to access the backing store, which is owned by root.
start scripts such as <code>/etc/init.d/iw.server</code> and the scripts in <code>iw-home/private/bin</code>	Needs to load the wfs kernel module.
<code>iw-home/private/bin/iwfsfix</code> <code>iw-home/private/bin/iwfsck</code>	Needs to access the backing store, which is owned by root.  These tools are intended only for use only on the advice of Interwoven Support or CSO staff. These tools are not part of the day-to-day operation of TeamSite.
<code>iw-home/bin/iwconvert</code> <code>iw-home/bin/iwmigrate</code> <code>iw-home/bin/iwcpwa</code> <code>iw-home/bin/iwcpfile</code>	Needs to access the backing store, which is owned by root.  These tools are related to the conversion of pre-5.5.1 backing stores to the new high-performance backing store format.
<code>iw-home/bin/iwwfconvert</code>	This tool converts in-process workflows from pre-5.5.2L versions of TeamSite.  It reads the source backing store (for which it needs to be root) and writes to the destination backing store.

<b>Read by iwserver</b>	<b>Comments</b>
All TeamSite configuration files	<code>iwserver</code> needs to be able to read TeamSite configuration files, but they can be owned by another user.

<b>Other</b>	<b>Comments</b>
iw-home/httpd/iw-bin/iwbs.cgi iw-home/httpd/iw-bin/iwbssdiff iw-home/httpd/iw-bin/iwbssdir iw-home/httpd/iw-bin/iwbssls iw-home/httpd/iw-bin/iwbspeek	<p>These backing store tools are not shipped with setuid turned on. However, these tools do need to be setuid root to operate properly.</p> <p>These tools are intended only for use only on the advice of Interwoven Support or CSO staff. These tools are not part of the day-to-day operation of TeamSite.</p>

Though most of the CGIs are installed by default with the read and write bits on, these CGIs do not necessarily need to be world-readable. Though some of these CGI programs are setuid root, they perform specific and limited functions. The CGIs contain no embedded shell interpreters or similar software that allows general operations.



# Index

---

## Symbols

.uid files  
about 33  
location of 84

## A

aborting server operations 119  
absolute paths 178  
access  
  files read by iwserver 350  
  privileges, to TeamSite 81  
  required for root 349  
  to files 82  
accessing TeamSite 65  
activating backing stores 264  
adding  
  custom menu items 134  
  metadata capture to the  
    TeamSite GUI 220  
  metadata search to the  
    TeamSite GUI 227  
  users to TeamSite 34, 43, 81, 84,  
    104  
admin.uid file 33  
administration GUI  
  about 99  
  and iw.cfg 99  
  and log files 100  
  applying settings 101  
  logging in 102  
  navigating 101

performing server  
  operations 100  
refreshing 102  
viewing system  
  information 99, 102

Administrators  
  abilities 93  
  about 83  
  defined 18

Apache  
  configuring 51  
  configuring MIME types 59  
  web server aliases 34, 43

application variables 288

area labels  
  configuring 111, 124

assigning files 18

attribute filtering 167

attributes  
  in LDAP schemas 150  
  of windows in custom menu  
    items 137

authentication  
  expiration 132  
  external file 146  
  LDAP 146  
  local 146  
  PAMs 146  
  password 81  
  roles 146  
  setting type 146

user 146  
users 146  
author.uid file 33

Authors  
  abilities 93  
  about 83  
  creating tasks 129  
  defined 18  
  editing files 129

Autoprivate  
  about 158  
  configuring 158  
  matching filenames 159  
  matching patterns 158

autoprivate.cfg  
  about 121, 280  
  format 158  
  location 158, 280  
  sample 160

## B

backing store  
  access control 274  
  activating 264  
  backing up 276  
  comments 263  
  converting active  
    workflows 252, 273  
  converting from command  
    line 256

creating new 242  
 disk space 26  
 encoding of names 262  
 freezing 167  
 location 29  
 moving 242, 265  
 multibyte characters 262  
 repairing 236  
 specifying location of 32, 41  
**backups**  
 multiple stores 276  
 of workareas 276  
 strategies 277  
**branch and workarea**  
 security 108, 154  
**branches**  
 creating 70  
 defined 15  
 locking models on 90  
 permissions 92  
 read access 154  
 remappings, configuring 180  
 restrictions on names 70  
 setting locking model 71  
 structure 243  
**browsers**  
 clearing cache 121  
 compatibility with TeamSite 28  
 requirements 27  
 windows, configuring 133  
**buttons, disabling** 140

**C**

cache size 115, 164  
 captured subexpression 290  
**Casual Contributor Interface**  
 about 129  
 expiring authentication 132

**Casual Contributor interface** 132  
**CGI scripts**  
 adding to the GUI 134  
 creating 134  
 in WebDesk Pro 136  
 internationalization 322  
 window attributes 137  
**CGI wrapper**  
 available variables 134  
**changing**  
 file attributes, on submission 167  
 group ownership of workareas 87  
 valid search paths 228  
**charset parameter, content**  
 encoding 323  
**checking**  
 disk space usage 243  
 for multiple servers 230  
 request handling 231  
 server status 230  
**chgrp command** 87  
**clients**  
 Microsoft network 66  
 NFS 68  
 TeamSite 65  
**CLTs**  
 internationalization 322  
**iwconvert** 256  
**iwfreeze** 260  
**iwidmap** 268, 274  
**iwmigrate** 270  
**iwreset** 263  
**iwstoreadm** 264  
**comments**  
 publish 79  
 submit

individual file 76  
 keywords 76  
 submit operation 76  
**comparing files**  
 viewing results 90  
**compressing**  
 editions 244  
**configuration files**  
 list of 279  
 locations 156  
 moving 156  
**configuration options**  
 available 121  
 requiring a restart 121  
**configuring**  
 area labels 111, 124  
 Autoprivate 158  
 backing store freezes 116, 167  
 branch and workarea security 108, 154  
 branch remappings 180  
 cache size 115, 164  
 CGI programs 56  
 custom menu items 134  
 default permissions on files 106, 107, 155  
 different web servers 189  
 directory operations 142  
 disabling Editor publish capability 108, 128  
 disk space 26  
 edition views 110, 126  
 encoding of text files 323  
 external remappings 190  
 file locations 156  
 file system active area cache 166

file system threadcount 115, 165  
group remapping 106, 107, 156  
history views 110, 127  
Home pages 127  
IP addresses 235  
iw-mount alias 52  
job attributes 144  
jobs in the GUI 110, 143  
launching files through  
  *iproxy* 163  
LDAP 148  
lock behavior 109, 154  
log files 116  
main branch locking  
  model 153  
main branch ownership 153  
main configuration file 121  
menu items 140  
metadata capture 200  
MIME types  
  Apache 59  
new browser windows 133  
PAM 151  
preview windows 110, 133  
proxy server 177  
RPC threadcount 116, 165  
rule sets for metadata  
  capture 205  
Samba 60  
Submit and Update logs 154  
Submit button 139  
submit filtering 167  
TeamSite administration  
  GUI 99  
  templates 161  
throughput monitors 116, 166  
user authentication 146

web server uid 152  
web servers 51, 111  
conflicting edits 90, 91  
connecting to TeamSite  
  FTP 28  
  conserving disk space 244  
  content-provider user role 19  
  content-provider.uid file 33  
  conventions, notation 11  
  converting active 252, 273  
  copying files 75  
  CPU requirements 24  
  creating  
    branches 70  
    new backing store 242  
    tasks, through WebDesk 129  
    workareas 73, 82  
custom menu items  
  adding to the GUI 134  
  CGI scripts 134, 136  
  HTML pages 138  
  window attributes 137  
custom scripts  
  creating 134

## D

data store 242  
database, LDAP 148  
*datacapture.cfg*  
  about 198  
  annotated example  
    database element 212  
    DATE datatype 213  
    instance 213  
    metadata identifier 212  
    rule identifier 212  
    UTF-8 encoding 212  
    validation-regex 213

configuring 205  
DTD 206  
location of 198  
debugging  
  proxy server configuration 194  
  submit filtering 171  
default  
  user authentication 146  
default file locations 29  
default permissions  
  on files 106, 107, 155  
deleting  
  branches 246  
  user name 34, 43  
directories  
  disabling operations on 142  
  permissions 93  
disabling  
  buttons 140  
  directory operations 142  
  menu items 140  
  SmartContext Editing 128  
  Submit button 139  
  unlocked file upload 109, 143  
disk space  
  checking usage 243  
  compression 244  
  conserving 244  
  data store 242  
  file system mount 243  
  low 167  
  managing 242  
  moving the backing store 245  
  partitions 26  
  recovery 244  
  removing old versions 245  
  requirements 24  
  usage 243

document root  
 configuring 180  
 defined 179  
 mapping 179  
**DTD**  
`datacapture.cfg` 206  
`metadata-rules.cfg` 201

## E

**editing**  
 files through WebDesk 129  
 text editor application 27  
**editions**  
 allowing Editors to publish 128  
 defined 17  
 initial 71  
 new 78  
 number of displayed 126  
 publishing 78  
*see also* publishing editions  
 viewing 110, 126  
 views, configuring 110, 126  
**editor.uid** file 33

**Editors**  
 abilities 93  
 about 83  
 defined 18  
 disabling publish  
     capability 108, 128

**email**  
 and tasks 145  
 mapping files 145  
 setting domains 145  
 setting servers 145  
 settings, for workflow 145  
**Embedded Failsafe** 195  
**enabling**  
     SmartContext Editing 128

    SmartContext QA 163  
**encoding**  
     charset parameter 323  
     `file_encoding.cfg` 297, 323  
     html files 283  
     IANA prefered names 297  
     Merge tool 297  
     META tag 323  
     of backing store names 262  
     of contents of `iw.cfg` 262  
     setting in `iw.cfg` 146  
     Source Differencing tool 297  
     specifying 323  
     text editors 263, 324  
     text files 283  
     Unicode 320  
     UTF-8 296, 320  
     valid charsets 297  
     visual differencing 297  
     vpaths 283  
**entity database** 127  
**environment variables,**  
     `LANG` 164  
**errata** 13  
**external remappings,**  
     configuring 190

**F**

**failover** *see proxy server*  
**file attributes**  
     changing on submission 167  
**file system**  
     active area cache 166  
     mount 243  
     threadcount 115, 165  
**file system interface**  
     and FTP 66  
     network connection 28  
**FTP**  
     using 28  
`file_encoding.cfg` 121, 283, 323  
     Merge tool 297  
     sample file 298  
     SmartContext Editing 296  
     Source Differencing tool 297  
     UTF-8 296  
     valid encodings 297  
     visual differencing 297  
**files** 234  
     access to 82  
     assigning 18  
     configuration 121, 283  
     configuring default  
         permissions 106  
     creating 161  
     default permissions 107, 155  
     encoding 121, 283  
     `file_encoding.cfg` 121, 283, 297,  
         323  
     merging  
         and Submit locking 90  
         and Write locking 91  
     OpenAPI configuration 234  
     `openapi.cfg` 234  
     parsing .shtml 57  
     permissions 93  
     private 158  
     setting permissions on 82  
     submitting to the staging  
         area 76  
     templates 161  
         virtual 244  
     filtering, on file submission 167  
     finding installation directory 232  
     freezing the backing store 119,  
         167

- and the file system interface 66  
clients, using 68  
connecting to the TeamSite server 28  
fully-qualified paths  
*see* proxy server
- G**
- generating new encryption keys 235  
Global Report Center  
installing 36, 45  
requirements 27  
groups  
creating 86  
files 85  
membership 86  
NFS limitations 156  
remapping 106, 107, 156  
GUI  
Casual Contributor 132
- H**
- hardware requirements 24  
High Availability  
about 301  
components 302  
configuring 301, 304  
installing 303  
iw.powerfail 303, 304  
iw.processfail 303, 305  
iwtcock 303  
logging 304, 305  
starting and stopping the server 306  
uninstalling 307  
history views  
configuring 110, 127
- Home page  
setting 127  
host headers, remapping  
*see also* proxy server  
host permissions  
setting 106  
HTML pages  
adding to the GUI 138  
httpd user name 52  
HTTPS requests  
redirecting 59
- I**
- IANA charset names 297  
in-context QA  
enabling 163  
initial edition 71  
inode count  
low 167  
inode requirements 26  
installation directory  
locating 232  
installation files 30, 38  
installation log file 30, 38  
installing  
adding users 34, 43  
license keys 47  
required access for 30, 38  
TeamSite 30, 38  
TeamSite Templating 23  
Intelligent File System  
about 21  
intermediate variables  
variables  
intermediate 288  
internationalization  
browser behavior 324  
CGI scripts 322
- CLTs 322  
encoding setting in iw.cfg 146  
file\_encoding.cfg 283  
IANA charset names 297  
iw.cfg settings 163  
recommendations 323  
regex\_map 296  
server\_locale setting 163  
SmartContext Editing 283  
text editor encoding 262, 324  
Unicode 320  
UTF-8 296, 320  
vpath encoding 283  
Internet Explorer  
compatibility 28  
Interwoven administration GUI  
*see* administration GUI  
Interwoven Merge tool 283  
invalidating user sessions 235  
IP addresses  
changing 235  
iPlanet web server  
configuring 51  
configuring aliases 60  
iw.cfg  
about 121  
activating change to 263  
and the administration GUI 99  
and the proxy server 179, 180  
authentication section 147  
configuration options 121  
encoding of 262  
encoding setting 146  
locating 49, 281  
metadata search paths in 228  
iwickrole 89  
iwconvert 256  
iwfreeze 260

iwgetelog 233  
 iwgethome 79  
 iwgetlocation 48  
 iw-home  
     creating 30, 38  
 iwidmap 268, 274  
 iwininstall 30, 38  
 iwmigrate 270  
 iw-mount alias  
     configuring 52  
 iwprefconv 127  
 iwproxy  
     configuring 177  
     debug option 194  
     launching files 163  
 iwreset 263  
 iwservletd 80  
 iwsessionkeygen 80, 235  
 iwgetstat 234  
 iw-store directory  
     backing up 50  
 iwsstoreadm 264  
 iwttemplates.cfg  
     about 121  
     format 161  
     location 161, 280  
 iwui user 32, 80  
 iwuinstall 79  
 iwwebd 80  
 iwwfconvert 252, 273

## J

Java  
     servlet engine 111, 152  
 JavaScript 64  
 jobs  
     attributes 144  
     defined 20

filters 144  
 listing in the GUI 110, 143  
 settings 144

**L**  
 labels, configuring 124  
 LANG environment variable 164  
 languages  
     browser behavior when  
         interpreting encoding 324  
 LaunchPad  
     about 65  
     applet 131  
     application 131  
     installing 65  
     localized GUI 322  
     setting server names 132  
     setting the default  
         interface 131  
 LDAP 146  
     and OpenAPI 148, 150  
     and operating system  
         authentication 150  
     configuring 148  
     database 148  
     modifying schemas 150  
     schemas 148  
     servers 146  
     TeamSite role  
         authentication 146  
     user authentication 146

license keys  
     format 47  
     generation page 47  
     installing 47  
     obtaining 47  
     troubleshooting 48  
     loading content 69

localized features 318  
 locales  
     native 163  
     TeamSite server setting 163  
 locating  
     installation directory 232  
 locations  
     of configuration files 156  
     of iw.cfg 49, 281  
     of roles files 281  
     of TeamSite files 156  
 locking  
     files, and uploading 109, 143  
     in workareas 90  
     Mandatory Write, defined 91  
     Optional Write, defined 91  
     Submit, defined 90  
     types of 90  
     Write, about 91  
 locking model  
     on the main branch 153  
     setting 71  
 locks  
     configuring behavior of 109,  
         154  
 log files  
     and the administration  
         GUI 100  
     configuring 116  
     reviewing 233  
     viewing 117  
         through the GUI 100  
 logging in  
     authentication 81  
     to the administration GUI 102  
 logging users out 235  
 login authentication expiration  
     default 132

setting 132  
login names 84  
login screen  
  selecting a GUI 65  
  selecting role 65  
logs  
  submit 154  
  update 154  
low disk space  
  detecting 116, 167  
low inode count  
  detecting 167

**M**

macro expansion, at submit time 172  
main branch  
  locking model 153  
  ownership 153  
managing server resources. *see* server resources  
Mandatory Write locking  
  defined 91  
master.uid file 33  
Masters  
  abilities 93  
  about 19, 83  
memory requirements 25  
menu items  
  disabling 140  
Merge tool 283, 297  
merging files  
  and Submit locking 90  
  and Write locking 91  
META tag  
  specifying web asset encoding 323  
metadata capture

about 197, 198  
adding to the TeamSite  
  GUI 220  
and DAS  
  synchronizing 226  
and workflow 222  
components 198  
configuration files 198  
configuring 200  
  appearance 198  
  names 198  
DTD 201, 205  
extended attributes 221  
form 205  
initiating from within a job 222  
required input 199  
results of 221  
rule sets 205  
schematic 199  
validating input 199

metadata search  
about 197, 223  
adding to the TeamSite  
  GUI 227  
and DataDeploy 224  
and iw.cfg 225  
and metadata capture 223  
changing valid search paths 228  
components 224  
configuration files 224  
configuring 226  
making fields non-searchable 228  
overview 223  
prerequisites 223

metadata-rules.cfg  
about 198  
configuring 201

DTD 201  
examples 201  
location of 198  
rule identifier 202  
UTF-8 encoding 202  
vpath identifier 202

MIME types  
  configuring  
    Apache 59

monitoring  
  system status 166

mounting the TeamSite server 231

moving  
  backing store 242  
  configuration files 156

moving backing stores 265

multibyte characters  
  browser behavior when interpreting encoding 324  
  in backing store names 262

multiple servers  
  checking 230

MultiStore  
  backing up 276  
  defined 15, 247

**N**

navigating through the administration GUI 101  
NCSA web server aliases 34  
NetBEUI 66  
Netscape browser  
  compatibility 28  
Netscape Enterprise Server  
  configuring 51  
  configuring aliases 60  
network drive 65

network file system 28

new users

  adding 81

NFS

  and the file system interface 28

  clients 65, 68

  group limitations 106, 107, 156

  server mount point 52

nobody 80

notation conventions 11

## O

od-admin user role 19

od-admin.uid file 33

od-user user role 19

od-user.uid file 33

OpenAPI

  and LDAP 148, 150

  configuration 234

  server 234

OpenAPI server

  about 234

  starting and stopping 235

  verifying 234

Optional Write locking

  defined 91

ownership

  of workareas, changing 87

## P

PAM

  and TeamSite 151

  configuring 151

  third-party modules 152

PAMs 146, 149

parsing .shtml files 57

passwords 82

  authentication 81

enabling plain-text 67

  setting 84

  Windows encrypted 67

paths

  absolute 178

  relative 178

  resolving *see also* proxy server

pcnfsd 28

  installing 35, 44

performance

  monitoring 166

permissions

  branch 81, 92

  directory 93

  file 81, 82, 93

  required for actions 91

  types of 91

  workarea 81, 92

port number

  proxy server 177

  servlet 152

  web server 177

  specifying 52

preview windows

  configuring number 110, 133

printing system information 103

private files 158

Professional Services 236

profiles, user 127

program files

  location 29

proxy server

  about 175

  configuring 175, 177

    basic operation 177

    mappings 112

    through the GUI 111

to use different

  webservers 189

debugging 194

document roots 181

external remappings 190

failover 192

fully-qualified paths 182

  client configuration 183

  configuring Internet

    Explorer 186

  configuring Netscape 183

  server configuration 182

host header remappings 191

host name 177

port number 177

redirecting TeamSite

  views 186, 187, 189

relative and absolute paths 178

remapping document roots

rules of precedence 175

SSI remapping 192

publishing editions

  about 78

  adding comments 79

Editors' ability to 18, 128

first edition 78

through the command line 79

through the TeamSite GUI 78

## R

RAID 0+1 26

RCS macro expansion

  about 172

  enabling 174

reconfiguring IP address 235

recovering disk space 244

redirecting HTTPS requests 59

redirecting TeamSite views *see also* proxy server

redirecting TeamSite views *see also* proxy server

redirector module 56

- and iwproxy 163
- for SSIs, installing 57

regex\_map

- element 286
- illustrated 284
- internationalization 296
- introduced 283
- regular expression syntax 287
- UTF-8 296
- variables 287

regex\_map element 283

regular expressions

- about 11, 168
- case-sensitivity 287
- expression engine 287
- file encoding 283
- in New File templates 161
- in regex\_maps 287
- in Submit filters 168

relative paths

- about 178
- see also* proxy server

remote contributors 175

removing

- menu items 140
- temp\_workareas 261
- users 84, 85, 104

repairing backing store 236

request handling

- checking 231

requirements

- backing store 26
- client 27

CPU 24

disk space 24, 27

hardware 24

inode 26

memory 25

resetting the TeamSite server 120

resolving path names *see also* proxy server

reviewing TeamSite logs

- overview 233

roles

- about 83
- adding users 104
- authenticating with LDAP 146
- TeamSite 81

roles files

- adding users to 84
- locating 281
- location of 84
- master users needed 52

root access

- about 349
- and shutdown operations 30, 38, 51
- and startup operations 30, 38, 51
- files executed by root 350
- files setuid root 349, 351
- gaining 30, 38, 51
- installing TeamSite 30, 38, 51
- required for TeamSite 349

RPC threadcount

- configuring 116, 165

rule sets

- configuring 205

**S**

Samba

about 60

configuring 60

connecting to the file system

- interface 28, 65

installing 35, 44

SCE

- see* Smart Context Editing

schemas, LDAP 148

SCSI controllers and drives 26

searching metadata

- components 224
- configuring 226
- overview 223
- prerequisites 223

second-predecessor links 261

security 81

server

- load, monitoring 234
- mount, verifying 231
- names, setting, for
- LaunchPad 132

operation, verifying 230

operations, aborting 119

operations, through the

- administration GUI 100, 119

resources

- disk space 242
- managing 242
- status, checking 230

server locales 163

server\_locale setting 163

servers

- LDAP 146
- starting and stopping 235

server-side includes 57

Service Packs, uninstalling 79

servlet engine 111, 152

- servlet port 152
  - SetHomePage feature 127
  - setting
    - Home pages 127
    - TeamSite permissions 107
  - settings
    - email, for workflow 145
    - encoding 146
    - server\_locale 163
  - SmartContext Editing
    - disabling 128
    - enabling 128
    - encoding of text files 283
    - file\_encoding.cfg 296
  - SmartContext QA
    - enabling 163
      - for SSIs 56
  - Source Differencing tool 283, 297
  - SSIs
    - enabling 56
    - requests, remapping and
      - virtualizing *see also* proxy server
  - SSL support 175
  - staging area
    - defined 16
    - on a new branch 71
  - starting TeamSite server 234
  - stopping TeamSite server 234
  - Submit and Update logs, size
    - of 154
  - Submit button
    - configuring 139
    - disabling 139
  - submit filtering
    - about 167
    - debugging 171
  - RCS macro expansion 172
  - enabling 174
  - sequence of events 169
  - when populating a workarea 75
  - Submit locking
    - defined 90
    - submitting files under 90
  - submit workflow process
    - and the Submit button 139
  - submit.cfg
    - about 121, 167
    - format of 168
    - location 167
    - sample 169
  - submitting
    - files, to the staging area 76
    - locked files 154
    - under Submit locking 90
    - under Write locking 91
  - submitting files, changing
    - attributes 167
  - synchronizing metadata capture and DAS 226
  - syndicate-admin user role 19
  - syndication-admin.uid file 33
  - system information
    - printing 103
    - viewing 99, 102
  - system services
    - iwprefconv 127
  - system status, monitoring 166
- T**
- task ownership 93
  - tasks
    - and email 145
    - defined 21
  - TeamSite
- accessing
    - through the file system 65
    - through the GUI 64
    - troubleshooting 66
  - adding metadata capture to the GUI 220
  - adding metadata search to the GUI 227
  - adding users 34, 43
  - administration GUI 99
  - architecture 21
  - clients 64, 65
  - configuration files 279
  - configuring
    - through the GUI 99
  - file locations
    - changing 156
  - High Availability 301
    - see also* High Availability
  - installation directory 30, 38
  - installing 30, 38
  - mounting 65
  - permissions, setting 107
  - populating 75
  - proxy server 175
    - configuring mappings 112
  - proxy server *see also* proxy server
  - proxy server, configuring 111
  - resetting the server 120
  - roles 81, 104
  - uninstalling 79
  - uninstalling Service Pack 1 79
  - upgrading 49
  - user roles 83, 104
  - users, adding and
    - removing 104
  - web daemon 175

TeamSite installation  
    CD-ROM 30, 38

TeamSite server  
    answering requests 231  
    freezing and unfreezing 119  
    mounting 231  
    process 230  
    restarting 230  
    starting 234  
    stopping 234

TeamSite Templating  
    installing 23  
    localized GUI 322

Technical Support 236

temp\_workareas 261

templates  
    configuring 161

text editor encodings 263, 324  
    encoding  
        text editors 296

throughput monitors 116, 166

troubleshooting  
    license keys 48  
    overview 236  
    repairing backing store 236

tslicinfo.log file 47

**U**

Unicode  
    about 320

uninstalling TeamSite 79

UNIX permissions  
    interaction with TeamSite 93

unlocked file upload,  
    disabling 109, 143

upgrading TeamSite 49

uploading files 109, 143

URL access 129

user authentication 146  
    re-encrypting 235

user profiles 127  
    about 127

user roles  
    about 83  
    Administrator 83  
    assigning 34, 43  
    Author 83  
    checking 89  
    content-provider 19  
    Editor 83  
    Master 83  
    od-admin 19  
    od-user 19  
    permitted actions 92, 93  
    roles files 33  
    syndicate-admin 19

user sessions  
    invalidating 235

users  
    adding 84  
    authenticating 146  
    authentication using LDAP 146  
    expiring authentication 132  
    iwui 32, 80  
    nobody 80  
    removing 84, 85

UTF-8  
    about 320  
    file\_encoding.cfg 296  
    recommendations 323  
    regex\_map 296

**V**

valid search paths  
    for metadata 228

variables

application 288

captured subexpression 290

naming convention 288

variables, regex\_map 287

verifying server mount 231

versions  
    number of displayed 127

viewing  
    branches and workareas 154  
    log files 117  
    system information 102

virtual files 244

vpaths  
    encoding mappings 283

**W**

web browsers  
    behavior when interpreting  
        encoding 324

requirements 27

web content  
    specifying the encoding of 323

web daemon  
    about 175  
    configuring 175, 177  
    setting defaults 152

web server  
    host name 177  
    port number 52, 177

web servers  
    aliases 29, 34, 43  
    configuring 51, 111, 189  
    iPlanet  
        configuring aliases 60

Netscape  
    configuring aliases 60

plugins 163, 177

starting and stopping 177

stopping and starting 59  
uid 106, 107, 152

**WebDesk**  
about 65  
expiring user  
    authentication 132  
localized GUI 322

**WebDesk Pro**  
about 65  
adding custom menu  
    items 136, 138

**window attributes**  
for custom menu items 137

**Windows networking** 28

**workareas**  
changing group ownership 87  
creating 73, 82  
defined 16  
locking files in 90  
permissions 92  
populating 75  
private 73  
read access 154  
shared 73  
submitting to the staging  
    area 76  
temp\_workareas 261

**workflow** 252, 273  
and metadata capture 222

**jobs**, defined 20

**models**  
    and jobs 20  
    defined 19

**tasks**  
    defined 21  
    specifying files in 222

**troubleshooting** 52

**Write locking**

## X

**XML**  
about 11  
datacapture.cfg 206  
metadata-rules.cfg 201  
regex\_map language 283  
special characters 293