



I N T E R W O V E N

Interwoven[®] Turbo Product Guide

V2.0

for IBM[®] Enterprise Information Portal and IBM Content Manager

© 2001 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 logo are registered trademarks of Interwoven, Inc., which may be registered in certain jurisdictions. SmartContext, DataDeploy, Content Express, TeamXpress, 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.



I N T E R W O V E N

Interwoven, Inc.

1195 West Fremont Ave.

Sunnyvale, CA 94087

<http://www.interwoven.com>

Printed in the United States of America

Version #2.0

Part # 40-10-40-42-04-200-602

Table of Contents

Chapter 1: About This Product 5

- What is an Interwoven Turbo Product? 5
- Features of the Turbo Product Family 6
 - Application Server Component 6
 - Personalization Server Component 6
 - Business-User Support 7
- Architectural Overview 7
 - General Turbo Architecture 7
 - Application Servers and Virtualization 9
- Who Can Use This Product? 10
 - Business Users 11
 - Application Developers 12
 - Presentation Layer Developers 12
- About This Document 13

Chapter 2: Introduction 15

- Overview 15
- Prerequisite Software 15
- Prerequisite Hardware 17
- Architectural Overview 18

Chapter 3: Installation and Configuration 21

- System Overview 21
- Installing Content Manager 22
 - Pre-Installation Procedure 22
 - Installation Procedures 22
 - Removing Content Manager 23
- Installing EIP on the WebSphere Application Server 24
 - Installing EIP 24
 - Removing EIP 25
- Installing the Turbo Components 26
 - Installation on the WebSphere Application Server 26
 - Installation on the TeamSite Server 27
 - Installation on the TeamSite Templating Client 29
- Setting Up Workareas 36



Chapter 4: Using This Product 39

Creating a DCR 39

Generating or Previewing DCRs with Presentation Templates 42

Customization 45

 Customizing DCTs 45

 Customizing Presentation Templates 50

 Customizing the RetrieveDoc Servlet 64

Chapter 5: Troubleshooting 69

Appendix A: Content Manager Sample Loader Program 71

Chapter 1

About This Product

What is an Interwoven Turbo Product?

Interwoven® Turbo is a solution software product that enables and accelerates the integration of Interwoven TeamSite® with a number of eBusiness suites. Each Turbo product provides a standardized set of integration points and baseline capabilities for a specific partner platform. However, Turbo does not restrict the customer to that single platform, but rather makes all of them available. Because of Turbo's modular approach and broad-based availability, TeamSite can be coupled with any of the leading eBusiness suites for which a Turbo has been developed.

The advantages of Interwoven Turbo are not limited to the flexibility and standardization it facilitates; Turbo also builds customer value by increasing Return On Investment (ROI) in the following ways:

- **Accelerates time-to-Web:** Turbo gets you up and running fast by eliminating the guesswork in getting enterprise products working together. Professional service expenditures can be targeted to adapting the solution to best meet the customer's environment rather than figuring out how to get the two products to work together.
- **Extends Investment:** As part of Interwoven's solution software, Turbo is licensed under a two-year subscription that entitles a customer to any Turbo upgrades released during the period.
- **Increases Productivity:** Turbo extends and enhances TeamSite's native capability to support all user types, from business users to Web and application code developers. It provides additional mechanisms for interacting with the eBusiness suite that help all contributors be more effective.
- **Content Reuse:** Interwoven Turbo will support a number of best-of-breed eBusiness suites so that customers can reuse content when they modify or build new applications.
- **Operational Stability:** Turbos, like all Interwoven products, are fully supported by Interwoven's enterprise-class technical support.

Features of the Turbo Product Family

Interwoven Turbo extends the functionality of TeamSite by providing components that seamlessly interface with eBusiness suite applications, such as Web application servers and personalization servers. Through these components, Turbo can provide an expanded set of capabilities to TeamSite users, including:

- Virtualization of dynamic Web applications during development
- TeamSite users can walk through an application in their workarea just like a customer
- Support for personalization servers through templating and metadata capture
- Workflow approval of content prior to submission
- Deployment of content (file system and database) to the production environment
- A simple business-user interface for contributing content to the Web properties

Application Server Component

“Out of the box” TeamSite virtualizes static Web content, which allows an entire Web site to be previewed from within a workarea while still in development. The Turbo application server component takes this one step further and provides the virtualization of dynamic content, such as JSPs, EJBs, ASPs, JHTML and servlets. This feature accelerates Web development by enabling in-context QA of the entire site from a workarea without disrupting the work of other contributors.

Additionally, Turbo usually provides an augmentation to the TeamSite User Interface (UI) to interact with the application server and register application components, and provide a mechanism for switching between “local” and “shared” server instances from within the TeamSite workarea.

Personalization Server Component

Turbo also includes a component for deploying template-entered data and metadata to repositories used by personalization servers and eCommerce applications. The dynamic targeting of specific content to particular users or groups can be based on comparisons of this data to the personalization schema constructed for a given business model. TeamSite supports this through Turbo by mapping metadata and the content personalization database so that content can be delivered using the correct rules. Furthermore, TeamSite stores the personalization rules and thus they can be versioned.

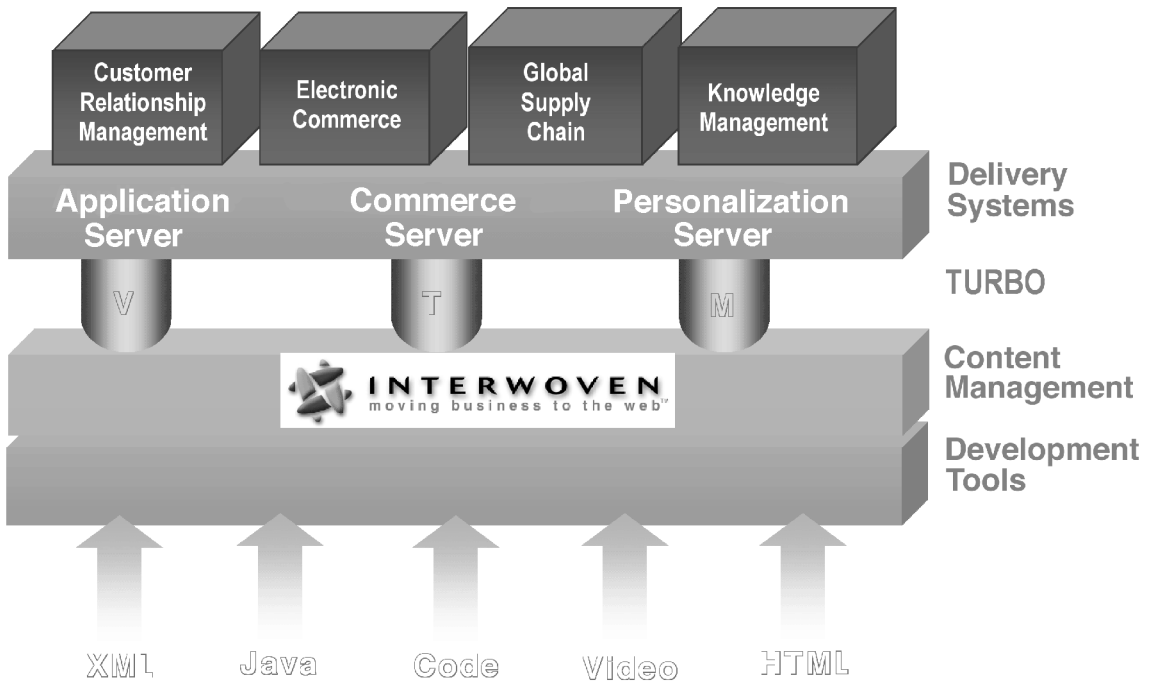
Business-User Support

Business users are the knowledge and process experts within a company. It is crucial that their content contribution be disseminated across the enterprise and effectively published to Web properties. Among the many ways in which Interwoven Turbo facilitates this are providing whole-site preview capability while still in development, updating repositories with content created in TeamSite templating or with specific metadata captured on Web site assets, and enforcing workflow approvals for content submission. Business-user support is a critical function of the Turbo product and will be discussed in greater detail below.

Architectural Overview

General Turbo Architecture

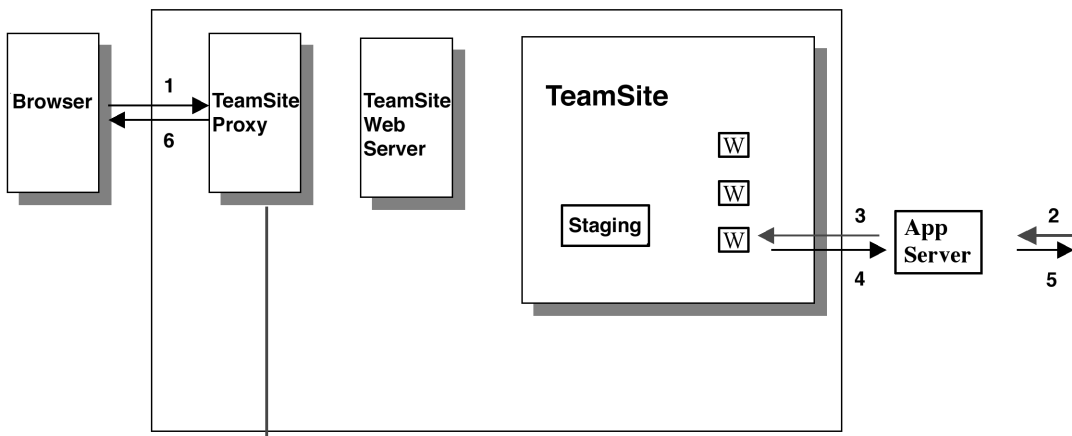
Turbo connects the TeamSite content management functionality to various delivery systems while augmenting the TeamSite native virtualization, templating and metadata capabilities. Development tools and delivery systems are all interconnected by the Turbo architecture. The result is that the customer's development environment is fully vertically integrated, from the initial back-end development of content to its deployment.



Turbo Architecture

Application Servers and Virtualization

The Turbo design is geared toward implementations with application servers. The TeamSite server, proxy server and Web server all interact with the application server to provide virtualization of dynamic, application-driven Web sites, as well as the native capability of TeamSite to virtualize static Web sites. At a high level, this is accomplished by configuring the TeamSite Proxy Server to redirect requests for certain file types to the application server and then pointing the application server toward the correct area of the originating request.

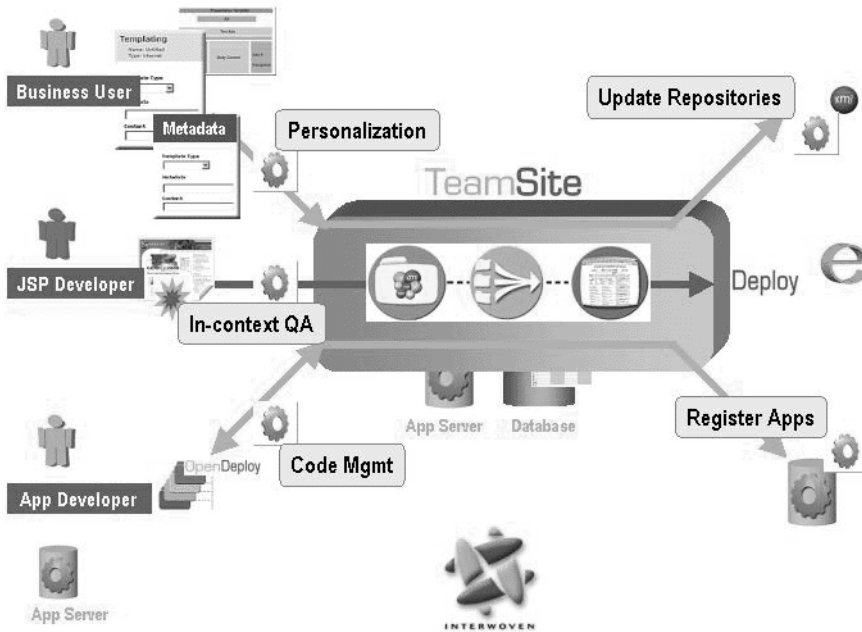


Application Server Integration

Instances of an application server can be set up to correspond either to a single TeamSite workarea or to be shared by multiple workareas, depending upon which types of users are involved. For example, presentation layer developers, such as content contributors and Web page designers, can share a single instance of the application server. On the other hand, Java™ developers should each have a dedicated instance (one application server instance per TeamSite workarea) because their work can affect the behavior of the application server. Turbo provides the flexibility to implement either of these architectures.

Who Can Use This Product?

Interwoven Turbo extends the inherent capability of TeamSite to support multiple types of users. As the diagram below shows, TeamSite with Turbo can connect back-end content contributors and their work to the databases, applications and repositories that power an enterprise-level Web site. The following section will describe three types of contributors—business users, application developers, and presentation-level developers—and discuss how they can benefit from the functionality provided by Turbo.



Interwoven Turbo Users

Business Users

Numerous Interwoven Turbo features let business users add their unique knowledge and expertise to Web properties. These features include templating, structured metadata capture coupled with personalization server support, and workflow approval.

Templating

Templating provides a means for the business user to contribute content through a simple, intuitive interface and then target that content to any number of repositories within the eBusiness platform. Templates are thus a means to capture the intellectual property of the business experts, who can also preview their content before submission. Specifically, the business users can:

- Enter structured content through a data capture template (DCT) based on a schema required by the personalization targeting rules.
- Preview content with presentation template applied to see how it will appear to the end user, and view dynamic content to test personalization rules.
- Publish content to Web properties through a structured and tailored mechanism.

Metadata Capture

Turbo lets business users:

- Apply metadata (tag) Web site assets based on a name/value pair specified in the repository schema and then ensure the deployment of these assets and metadata to the repository.
- Manage the development of and version control of personalization server rules.

Workflow Approval

The use of workflows to approve content before it is submitted provides safeguards and enforces business process.

Application Developers

Application developers develop and maintain the “application” logic and code for an application server. They are primarily concerned with the coding of the business logic and rules and managing access to databases and legacy applications. Turbo helps them through server selection, component registration, and site virtualization.

Turbo benefits application developers with:

- Simplified administration—Full source control for applications using a single TeamSite server.
- Parallel development—Developers can work in multiple workareas creating different pieces of the same application without conflicts.
- Simplified, in-context QA—Developers can preview the entire Web site from within a single workarea.
- Full control—Versioning, workflow, and rollback capabilities provided by TeamSite.
- Distributed development—Local or remote team development.
- Direct access for JSP files—Developers can preview JSP files directly from the TeamSite browser interface.
- Managed deployment of J2EE entities—A UI for managing deployment of J2EE entities to the application server and then for switching between server instances based on the environment.

Presentation Layer Developers

Presentation-layer developers are the content contributors and Web developers who create and modify JSP tags, JavaScript, HTML, DHTML, images and other site elements that will appear at the presentation layer. They are mainly concerned with the layout, design, and functionality of the Web pages. Interwoven Turbo provides support to presentation layer developers by providing in-context QA of dynamic Web sites. This can be done while the site is still in development, thus the development cycle is accelerated.

About This Document

The following Turbo Product Guide discusses the solution and implementation of TeamSite with one of the major eBusiness platform servers. It contains a description of the installation, configuration and implementation of the specific solution that can be used to get the suite component up and running with full TeamSite functionality.

Although this document contains general information about Turbo products that may be similar to other platforms and servers, for detailed information consult the product guide about the specific Turbo that will be implemented.

Note: When describing the installation and configuration of the Turbo product, this document refers to *TeamSite*. If your license is for the *TeamXpress*[™] product instead, the information contained in this document still applies to your product with the possible exception of branch creation. For further details, see the *Interwoven TeamXpress for Multiplatforms, WebSphere Edition Administration Guide*.



INTERWOVEN

Chapter 2

Introduction

Overview

This document describes the installation and configuration of Interwoven Turbo for IBM® Enterprise Information Portal (EIP) and Content Manager. This Turbo integrates TeamSite 4.5.1 with the IBM Enterprise Information Portal Version 7.1.2 and IBM Content Manager Version 7.1.

This product guide does not cover the full installation of TeamSite, EIP, or Content Manager, nor does it cover the usage of these products. It is expected that users and administrators of these software packages are familiar with installing and using TeamSite, EIP, and Content Manager.

Note: Installation of the EIP and Content Manager Turbo components requires that the WebSphere Application Server is already installed. For more information on this, refer to the *Interwoven Turbo Product Guide for IBM WebSphere Application Server*.

Prerequisite Software

For the the Turbo product for EIP and Content Manager to be installed, certain software components first must be installed for EIP, Content Manager and the Turbo itself.

- The Turbo product requires the following software components:
 - Windows NT® or Windows® 2000
 - TeamSite 4.5.1
 - TeamSite Templating 4.5.1
 - TeamSite Templating Java Client
 - IBM WebSphere® Application Server 3.5.3 (release 3.5 + fix pack 3)
 - One additional eFix for fix pack 3 - PQ46019 (either download from the eFix site or request through IBM support)
 - Enterprise Information Portal V 7.1.2 and Content Manager V7.1.0



- Database for the WebSphere Application Server
- Web server for TeamSite (IIS 4.0, NES 3.6.3 or Apache for Solaris™ is recommended)
- Web server for WebSphere Application Server
- The Templating Client needs Java UI support and JDK 1.2.2 or higher.
- In addition, EIP requires that you have the following software components installed:
 - Database (usually IBM DB2 UDB, which is included with EIP)
 - Java 2 SDK (included with EIP)
 - Browser version that supports frames and cascading style sheets

Note: Some features of EIP require additional software components; for the full set of software requirements for EIP, refer to the *Planning and Installing Enterprise Information Portal* manual.

- Content Manager requires that you have the following software components installed:
 - Database and toolkits
 - Java 2 SDK (included with Content Manager)
 - Browser version that supports frames and cascading style sheets
 - Microsoft Visual C++ Version 6
 - Install the Content Manager fix associated with APAR IR 44982 if you plan to use the IBM Video Charger with Content Manager.

Note: Some features of Content Manager require additional software components; for the full set of software requirements for Content Manager, refer to the *Planning and Installing Content Manager* manual.

Prerequisite Hardware

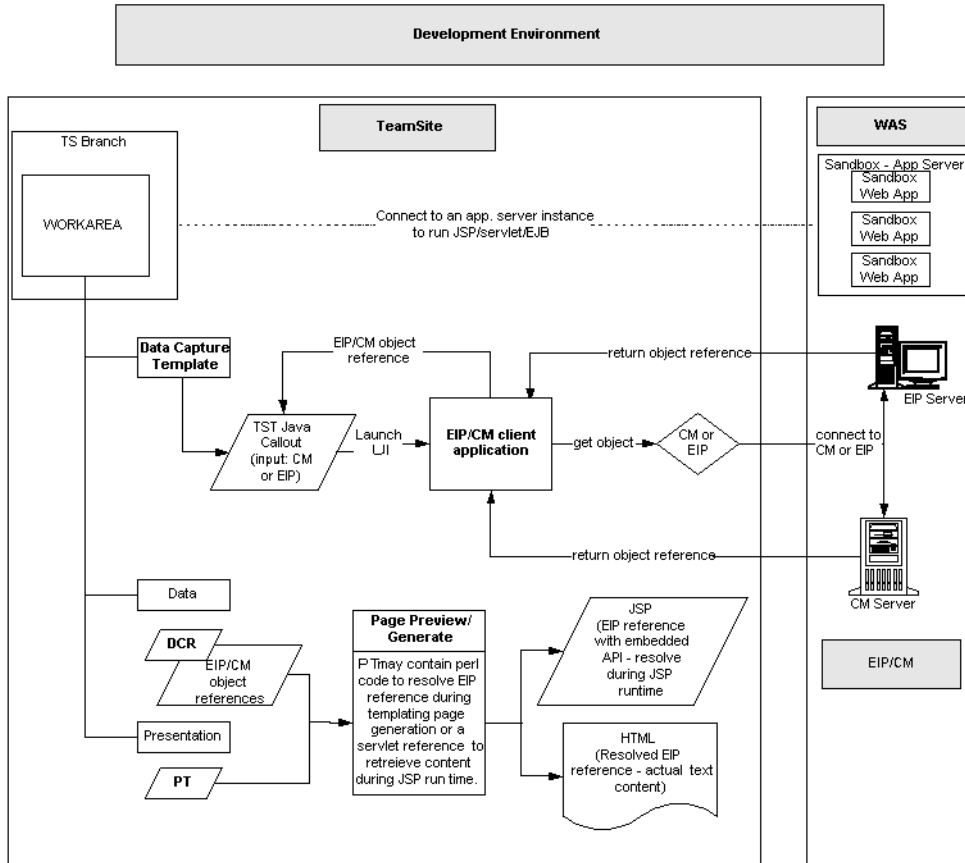
For TeamSite server configuration requirements, see the recommendations in the *Administering TeamSite* manual.

For hardware requirements for the EIP server, see the *Planning and Installing Enterprise Information Portal* manual.

For hardware requirements for the Content Manager server, see the *Planning and Installing Content Manager* manual.

Architectural Overview

The architecture of Interwoven Turbo for EIP and Content Manager is shown in the following diagram:



Turbo Architecture

The Turbo product enables TeamSite users to retrieve and select assets stored in IBM Content Manager or other data stores managed by EIP and to store these asset references into a DCR during the content entry process. They are embedded as DCR values through the EIP search GUI launched by the templating callout facility. Assets can be images, video files, audio files, or text files.

Some sample presentation templates that come with Turbo use Perl function calls to resolve EIP references and retrieve the asset first, then include the actual content of the asset from repositories outside the TeamSite environment during templating page generation process. This scenario is limited to pure ASCII text files only.

Other presentation templates will generate a reference URL to a servlet for content retrieval during JSP runtime (JSPs with embedded EIP references are generated by TeamSite Templating). In other words, if users have a virtualized workarea through the WebSphere Application Server Turbo component, they will be able to preview any JSP page that includes the EIP reference content.

Content Manager is the datastore for a wide variety of content types. You can connect to Content Manager directly through EIP or as part of a *federated* search for searching and retrieving content.

The typical setup runs TeamSite, EIP Client Kit, and Content Manager and WebSphere Application Server on separate servers.



INTERWOVEN

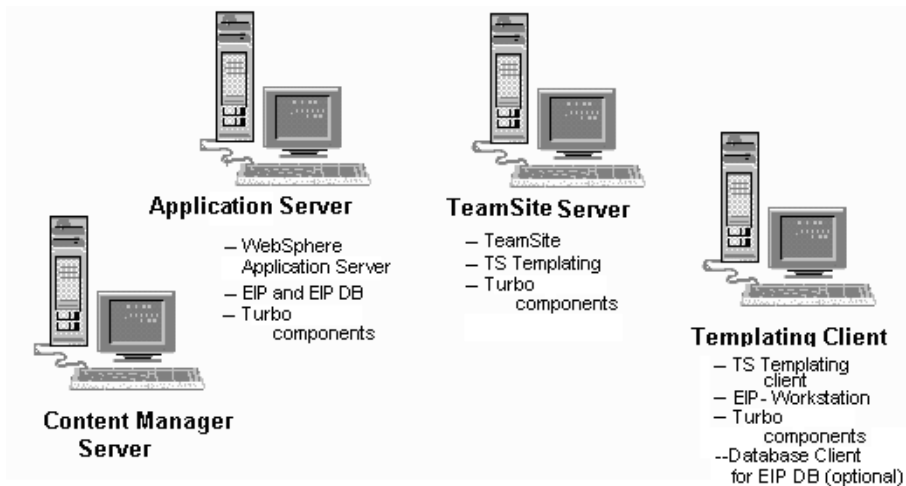
Chapter 3

Installation and Configuration

The installation procedure for Turbo 2.0 for IBM EIP and Content Manager consists of first installing EIP and Content Manager and then also installing Turbo components on the WebSphere Application Server, on the TeamSite server, and on the TeamSite Templating client. The administrator installing these components must have write-level file permissions for the directories in which they will be placed. It is recommended that the components be installed on separate servers.

System Overview

A typical system configuration is shown in the following diagram:



System Configuration

Installing Content Manager

The installation procedure here is intended for an initial installation of a basic Content Manager system on a workstation. For complete information on installing Content Manager, see the *Planning and Installing Content Manager* manual supplied with Content Manager.

Pre-Installation Procedure

Before you begin the Content Manager installation, ensure that you have the following items and information:

- All of the memory, hardware, and software requirements to install Content Manager.
- A user account that can be used to perform the installation. The account must be local and belong to the `Administrator` group. The user ID must follow the requirements for your relational database.
- A relational database; IBM DB2 UDB is supplied with Content Manager.
- Microsoft Visual C++. Ensure that the Visual C++ environment variable and program are configured for the Content Manager administrator's user ID.

Installation Procedures

1. Ensure that you are logged in using a User ID with `administrator` privileges.
2. Start the Content Manager installation from the CD by entering the following at a DOS prompt:
`x:\win\lang_code\setup.`
Where `x` is the local CD drive and `lang_code` is the directory for your language. For English, the directory is `ENU`.
3. Select the installation program language.
4. Select the Content Manager components you want to install. At a minimum, install the `Library server`, `Object server`, and `System Administration` components.
5. Set up the configuration panel for installation.
 - On the **Client/Server Configuration** tab: Specify the database you have and make sure that `Library Server` and `Object Server` databases are checked. Check to use a network table file.

- On the **Library Server Database** tab: Specify the drive for the database and accept the other defaults.
 - On the **Object Server Database** tab: Specify the drive and accept the defaults.
 - If you are using DB2 UDB, on the Windows Service Configuration, specify the password for the user ID for your databases.
 - On the **Network Table Generation** tab: Accept the defaults.
 - On the **System Administration Gateway Configuration** tab: Accept the defaults.
6. After completing the configuration panels, click **Next** on the Start Copying Files window to install Content Manager.
 7. After installing Content Manager, install the Content Manager Client for Windows to use with your Content Manager system. You should install the Client Application and Common Files component. Install the Toolkit and DBCS files if you need them. Follow the installation on the installation screens to complete the install.
 8. After successfully installing, you are ready to set up your Content Manager system and begin storing the documents and other content data.

Removing Content Manager

See the *Planning and Installing Content Manager* manual for information on removing Content Manager from your system.

Installing EIP on the WebSphere Application Server

On the WebSphere Application Server computer, you should install the following components:

- WebSphere Application Server
- Enterprise Information Portal
- Database (DB2 UDB for EIP)

This installation procedure assumes that you have installed the WebSphere Application Server with the appropriate updates; see the *Interwoven Turbo Product Guide for IBM WebSphere Application Server* for more information.

Ensure that the Y drive is mapped to the TeamSite iwserver share, `\\ts_host_name\iwserver`, where `ts_host_name` points to the location of TeamSite.

Installing EIP

The installation of the EIP server is described in the *Planning and Installing Enterprise Information Portal* manual. When installing on the WebSphere Application Server computer, note the following:

- Select the **EIP Development Workstation** button for the type of machine to install. This is the third option on the EIP installation screen.
- Check the **EIP Administration** check box in the component list, then check the **Admin Client** and **Admin Database** check boxes to install them.
- Similarly, select the following connectors to install .
 - Federated connector
 - Content Manager connector
- In the Network Table Generation dialog box specify the server names, ports and host names for Content Manager. If your Content Manager is on a remote system, you must supply these values.
- You can select the defaults for the other installation options. If you plan to connect to an existing text search or image search system, enter the information for those systems on the appropriate installation screens.

- If you chose to generate the Network Table (`FRNOLINT.TBL`) later, you can use the following commands to do so:
 1. From a command-prompt window, change to the directory where Content Manager is installed, default is `frnroot`.
 2. Type **frnnlinc**
 3. Type **1** to add a server entry.
 4. At the `Select Server Location (Current:NONE)` prompt that appears, type **2** to add a remote server.
 5. At the `Select type of Server (Current:NONE)` prompt that appears, type **1** to select library server.
 6. At the `Select platform type (Current:NONE)` prompt that appears, type **3** to select Windows NT.
 7. Enter the server name, port number, and host machine address in the sequence and save.

Note: Users can modify the default content/MIME associations in the `cmbcc2mime.ini` file located in the EIP root directory (default is `C:\CMBROOT`). If the file is changed, it must be copied to the `\TemplatingClient` directory for the updates to be used when the content is viewed.

Removing EIP

See the *Planning and Installing Enterprise Information Portal* manual for information on removing Content Manager from your system.

Installing the Turbo Components

After you have installed Content Manager and EIP, you are ready to install the Turbo Integration Components.

Installation on the WebSphere Application Server

Before starting the installation, go to **Control Panel > System > Environment** and set the following system variables

`JAVA_HOME` set this environment variable to point to the location of the JDK. It is recommended to point to the JDK that is available with WebSphere, for example, `d:\websphere\appserver\jdk`.

`WAS_HOME` set this to point to the location of the application server directory for WebSphere, for example, `d:\websphere\appserver`.

Installation

As you work through the installation program, the defaults for the required information are displayed in square brackets []. You can accept the defaults, where appropriate, by pressing **Enter**. If the default value is not correct for your system, enter the correct information. You will also need to know the TeamSite branch and workarea to use for your Web pages.

Follow these steps to install the TeamSite Integration Components:

1. In a DOS-prompt window, run `install.bat` from the `webSphere` subdirectory of the installation program directory.
2. Enter **y** to install the WebSphere Application Server component of the Turbo. Refer to the *Interwoven Turbo Product Guide for IBM WebSphere Application Server* for more details.
3. Enter **y** to install the EIP component of the Turbo.
4. Enter the path for the `appserver` directory for WebSphere; for example, `X:\websphere\appserver`.
5. Enter the type of searches you want users to perform:
 - **Fed** for EIP federated searches

- **cm** for directly searching Content Manager (this is the default)
- 6. Enter the path for the location of EIP (default is `x:\cmbroot`). This is the home directory where you installed EIP.
- 7. Enter the name of the Content Manager library server or accept the default.
- 8. Enter the user ID for the Content Manager.
- 9. Enter the password for the Content Manager user ID you just entered.

After you enter this information, the installation program starts copying files for the Turbo components.

- 10. When the installation program prompts you whether to continue with the installation of WebSphere Personalization, enter **y/n**, depending on your requirements. Refer to the *Interwoven Turbo Product Guide for IBM WebSphere Personalization*.
- 11. Enter the number of lightweight sandboxes that you want to create.
- 12. Enter the number of heavyweight sandboxes you want to create.

After the sandboxes are created, they are equipped with the configuration (servlet and classpaths) required to operate with the Turbo components.

Uninstalling the Integration Components

In a DOS-prompt window, run `uninstall.bat` from the WebSphere subdirectory of the installation program directory.

Enter **n** to uninstall all the components.

Enter **y** to uninstall the EIP component of the Turbo, then enter the required information as prompted.

Installation on the TeamSite Server

Before starting the installation, go to **Control Panel > System > Environment** and set the following system variables:

`JAVA_HOME` set this to point to the location of the JDK (JDK1.2.2 or above).

`TS_HOME` set this to point to the TeamSite home directory, usually `c:\iw-home`.

Also, you will need to know the TeamSite branch and workarea your users will use when creating their Web pages.

Installing the Integration Components

Follow these steps to install the TeamSite Integration Components on the TeamSite server:

1. In a DOS-prompt window, run `install.bat` from the TeamSite subdirectory of the installation program directory.
2. Enter `y` to install the WebSphere Application Server component of the Turbo. Refer to the *Interwoven Turbo Product Guide for IBM WebSphere Application Server* for more details..
3. Enter `y` to install the EIP component of the Turbo.
4. Enter the TeamSite installation home, default is `c:\iw-home`.
5. Enter the host name or IP address of the WebSphere Application Server.
6. Enter the port for the WebSphere Application Server or accept the default, which is 80.
7. Enter the TeamSite branch name and workarea you want to use for samples and properties.

After you enter this information, the installation program starts copying files for the Turbo Components.

When the installation program prompts you whether to continue with the installation of WebSphere Personalization, enter `y/n`, depending on your requirements. Refer to the *Interwoven Turbo Product Guide for IBM WebSphere Personalization*.

This completes installation on the TeamSite server.

Uninstalling the Integration Components

In a DOS-prompt window, run `uninstall.bat` from the TeamSite subdirectory of the installation program directory.

Enter **n** to uninstall all the components.

Enter **y** to uninstall the EIP component of the Turbo, then enter the required information as prompted.

Installation on the TeamSite Templating Client

After you have installed TeamSite Templating Client and the DB2 UDB Client on the TeamSite Templating client computer, you should install the following components:

- Enterprise Information Portal–Workstation for developers
- Database client support (connection)
- Turbo components

Also, the TeamSite client display must be set to support at least 256 colors

Installing EIP—Workstation

When installing EIP–Workstation, note the following:

- Choose EIP Development Workstation as the type of computer to install.
- Select the following connectors to install:
 - Federated
 - Content Manager

In the Network Table Generation dialog box, specify the server names, ports and host names for Content Manager. If your Content Manager is on a remote system, you must supply these values.

You can select the defaults for the other installation options. If you plan to connect to an existing text search or image search system, enter the information for those systems on the appropriate installation screens.

If you are using EIP server, you will also want to define the federated entities and attributes for searching and create the search templates that your Web page authors will use for federated searches.

Removing EIP—Workstation

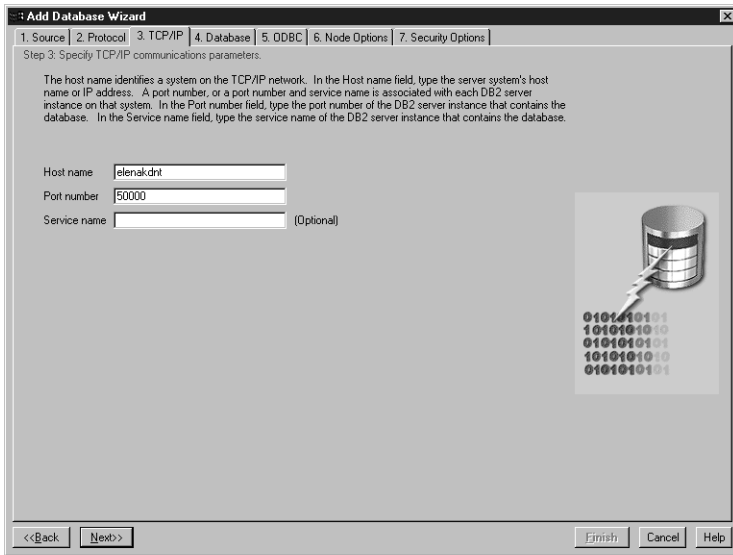
See the *Planning and Installing Enterprise Information Portal* manual for information on removing EIP from your system

Creating a Database Client Connection for the EIP Database

You must have a connection to the EIP database on the client computer if you are going to use federated searches. Make sure that your DB2 client is configured so that EIP client has access to the EIP database. Use the DB2 Client Configuration Assistant and the Connection Wizard to configure the connection.

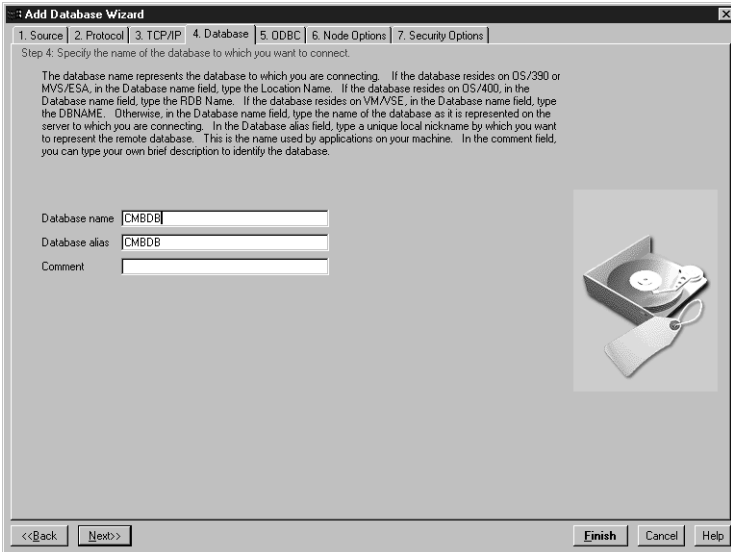
Select the EIP database and add a connection.

1. From the Client Configuration Assistant window, click the **Add** button to add a new database.
2. In the next window that opens (the **Source** tab of the Add Database Wizard), choose the option `Manually configure a connection to a database`, then click the **Next** button.
3. This will open the **Protocol** tab of the Add Database Wizard. Select the **TCP/IP** button and then click **Next**.



Add Database Wizard — TCP/IP Tab

4. The **TCP/IP** tab appears. Enter the Host name and Port number (and, optionally, the Service name) of the DB2 server in the fields provided. Click **Next**.



Add Database Wizard — Database Tab

5. The **Database** tab appears. Enter the Database name and Database alias (and, optionally, a comment) of the DB2 server in the fields provided. Click **Finish** to complete the setup.

After you set up the connection to the EIP database, update the classpath in TeamSite Templating Client file for the DB2 Client.

1. Open the file `tst.lax` in a text editor such as Notepad.

2. Find the line that defines the lax class path:

```
lax.class.path=../TemplatingClient/regex.jar;../TemplatingClient/
xerces.jar;../TemplatingClient/JConfig.zip;../TemplatingClient/
ExtDirpane.jar;../TemplatingClient/collections.jar;../TemplatingClient/
swingall.jar;../TemplatingClient;../TemplatingClient/tst45.jar;lax.jar;C:/
Program Files/Interwoven/TemplatingClient/eipcm/eipcm.jar;C:/cmbroot/lib/
cmbod71.jar;C:/cmbroot/lib/cmbview71.jar;C:/cmbroot/samples/java/client/
cmbclient71.jar;C:/cmbroot/lib/cmb71.jar;C:/cmbroot/lib/cmbsdk71.jar
```

3. At the end of this line, add the path for `db2java.zip`; for example, if the DB2 client is installed on the C: drive, add:

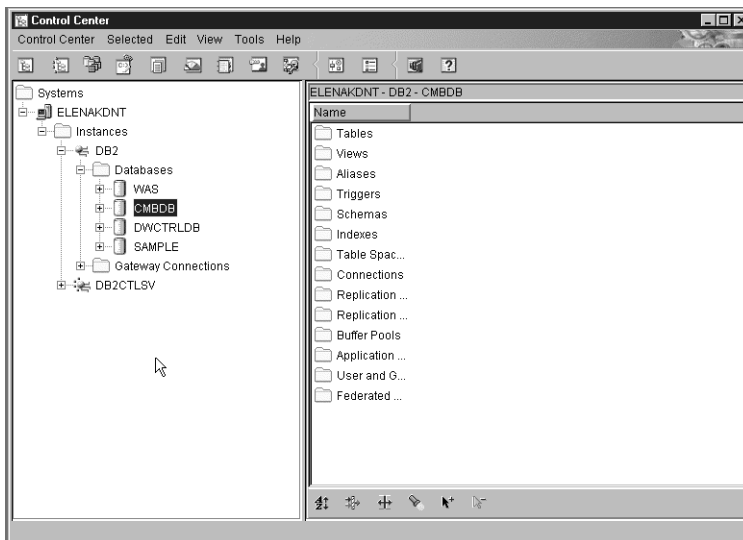
```
C:/Program Files/sqlllib/java/db2java.zip
```


Ensure that you have a semicolon between the last entry in the lax class path and the path you are adding.

4. Save the file and exit the text editor.

The user id with which the templating client workstation is logged on has to be added to the DB2 users of the database for Federated search (for example, CMBDB) on the DB2 server.

1. Bring up the DB2 control center from the **Start > Programs > DB2 .**
2. Go to the *CMBDB* node as shown below.



DB2 Control Center Window

3. Right-click on *CMBDB* and a context menu will appear. Select `authorities` from that menu. You will see the Database Authorities dialog box.



Database Authorities Dialog Box

5. Click the **Add User** button.
6. Add the user you want and give the user appropriate permissions by highlighting the added user and checking on the required permissions.
7. Close all DB2 windows.

Note: After you have created the EIP database client, before it can connect to the EIP database, you must ensure that the DB2 server's Authentication Type is set to `Client`.

Installing the Turbo Components

Before starting the installation, go to **Control Panel > System > Environment** and set the following system variable:

`JAVA_HOME` set this to point to the location of the JDK that comes with EIP: `CMBROOT/jdk/jre`; where `CMBROOT` is the root directory of the EIP installation.

After you have installed EIP, you are ready to install the Turbo components on your TeamSite Templating client computer.

Follow these steps to install the Turbo components on the TeamSite Templating Client computer:

1. In a DOS-prompt window, run `install.bat` from the `iweipclient` subdirectory of the installation programs directory.
2. Enter the directory where TeamSite Templating Client is installed or accept the default.
3. Enter the directory where EIP is installed.
4. Enter the type of searches you want your users to perform:
 - **Fed** for EIP federated searches
 - **cm** for directly searching Content Manager

This entry must match the types of searches you chose when installing the Turbo components on the Application server.

5. Enter the name of the EIP database when asked for the server name.
6. Enter the EIP user ID that will be accessing the EIP database.
7. Enter the password for the EIP user ID.

After you enter this information, the installation program starts copying files for the Turbo components.

When you have successfully installed the Turbo components, you can begin to use TeamSite Templating to create Data Content Records and preview the Web pages. Chapter 4, “Using This Product,” provides information on using TeamSite Templating, EIP, and Content Manager to add the content stored in Content Manager to Web pages.

Uninstalling the Turbo Components

In a DOS-prompt window, run `uninstall.bat` from the `iweipclient` subdirectory of the installation program directory.

To uninstall EIP components, type in the required information as prompted.

Setting Up Workareas

After completing the installation of the EIP, Content Manager, and the Turbo components, JSP pages cannot be previewed unless the workarea (that was specified in the Turbo installation) is connected to a WebSphere Application Server runtime environment. This is accomplished by allocating a heavyweight sandbox to the workarea. Users must use heavyweight sandboxes to preview JSPs and generated HTML page with EIP content reference.

To allocate a heavyweight sandbox to the workarea:

1. Log on to TeamSite and open the TeamSite UI.
2. Go into the workarea that user specified during installation; this should contain a file named `eip_config.xml`.
3. Open `eip_config.xml` and search for `[cmbroot]` and replace this string with the actual location of the EIP server `lib` directory (example `C:\cmbroot\lib` on the WebSphere Application Server computer). This config file contains a special servlet named `RetrieveDoc` to handle rendering content from Content Manager through an EIP federated search connector or Content Manager direct connector. For more information on how to use the `config.xml` file, refer to the *IBM WebSphere V3.5 Handbook* or the *Interwoven Turbo Product Guide for IBM WebSphere Application Server*.
4. Select **File>Sandbox Pool**. A sandbox selection form will be displayed.
5. Choose the heavyweight sandbox and specify the `eip_config.xml` and fill in the servlet classpath that contains any other servlet that will be used within your workarea.
6. Click **Check Out**. If successful, a status page will be displayed; if not, a status page with a specific error will be displayed.

Heavyweight sandbox users can develop their own `config.xml` to include other properties to configure their Web application. But the `RetrieveDoc` and the `Auto-Invoker` servlets are required to be part of the Web application. Without the `Auto-Invoker`, the `RetrieveDoc` servlet will not be able to call other servlets because these servlets are not registered with the Web application. For details how to construct `config.xml`, refer to *IBM WebSphere V3.5 Handbook*. Here is the XML to set up the `Auto-Invoker` and `RetrieveDoc` servlets:

```
<servlet name="RetrieveDoc" action="create">
<description>RetrieveDoc</description>
<code>RetrieveDoc</code>
  <init-parameters>
    <parameter name="propFilePath" value="[cmbroot]"/>
  </init-parameters>
<load-at-startup>true</load-at-startup>
<debug-mode>>false</debug-mode>
<uri-paths>
  <uri value="/RetrieveDoc"/>
</uri-paths>
<enabled>true</enabled>
</servlet>

<servlet name="Auto-Invoker" action="create">
<description>Auto-Generated - Serves Servlets by Classname</description>
  <code>com.ibm.servlet.engine.webapp.InvokerServlet</code>
<init-parameters/>
<load-at-startup>true</load-at-startup>
<debug-mode>>false</debug-mode>
<uri-paths>
<uri value="/servlet"/>
</uri-paths>
<enabled>true</enabled>
</servlet>
```


Chapter 4

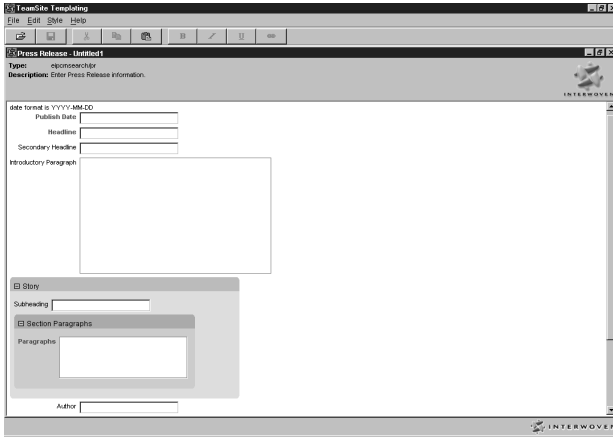
Using This Product

This chapter outlines how to use the Interwoven Turbo for EIP and Content Manager. It describes creating data content records (DCRs), generating and previewing DCRs using different presentation templates (PT), and customization. The Turbo installation copies four sample presentation templates (`pr1`, `pr2`, `pr3`, and `pr4`) into the presentation subdirectory of the workarea specified during installation. The use of these PTs is also covered in this chapter.

Creating a DCR

The Turbo product can be used to create data content records (DCRs) by executing the following steps:

1. Log on to the TeamSite server and go into the workarea specified during the Turbo installation.
2. From the File Menu, select the `New Data Record` option. In the window that appears, select `eipcmsearch`.
3. Then in the next window, choose `pr` and click the **Select** button. This will bring up the data capture template to enter data. Fill the form with relevant data. This form includes the **Text Item** and **Image Item** fields with **Search** buttons attached. These fields allow you to search EIP or Content Manager and bring the reference of the selected item back to the DCR.

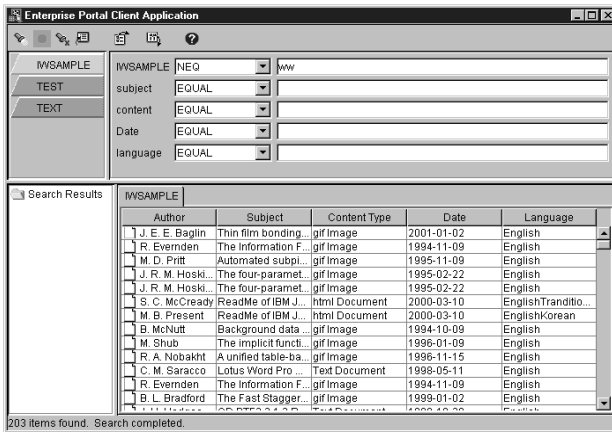


The screenshot shows a web browser window titled "TeamSite Templating" with a menu bar (File, Edit, Site, Help) and a toolbar. The main content area is titled "Press Release - Untitled1" and contains the following fields and sections:

- Type: eipnewschdr
- Description: Enter Press Release Information.
- Date format is YYYY-MM-DD
- Publish Date:
- Headline:
- Secondary Headline:
- Introductory Paragraph:
- Body:
- Section Paragraphs:
- Author:

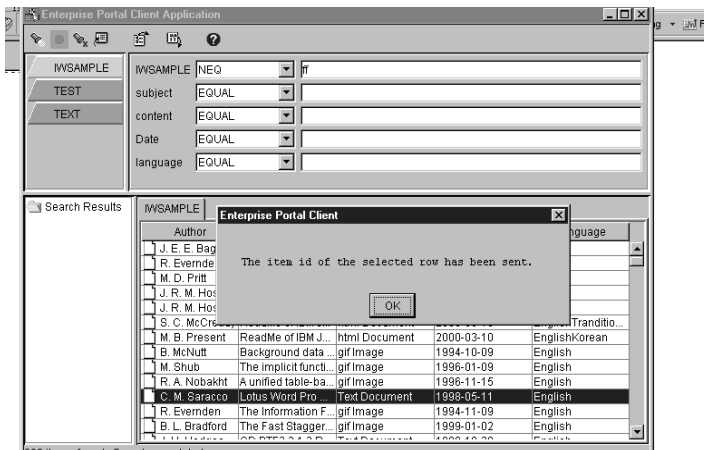
TeamSite Templating Data Capture Form

4. Clicking on the **Search** button for **Text Item Field** will launch an external Java application for EIP and Content Manager, the Enterprise Portal Client Search Window, which allows you to search for any text document item by specifying the search criteria. It is assumed that users will select the appropriate type of text document for content retrieval depending on how they plan to use this text document item (constraints applied to use with different PTs).



Enterprise Portal Client—Search Window

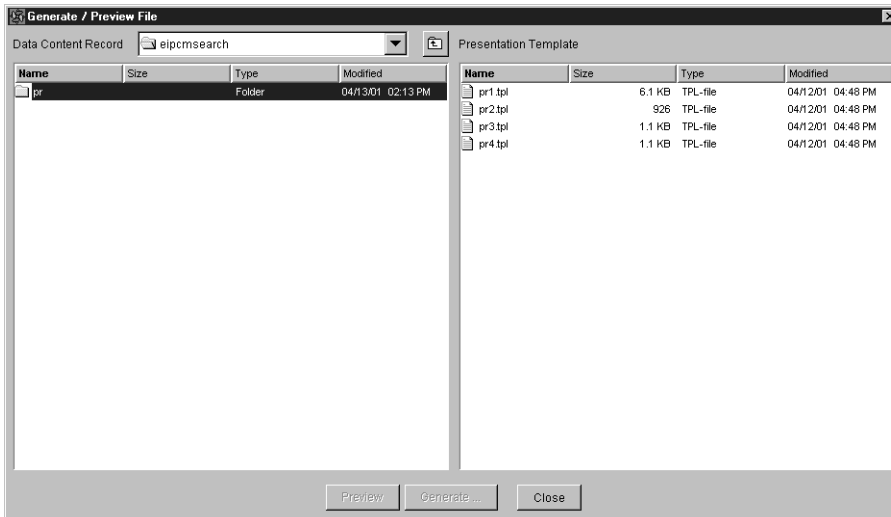
5. Click the **Search** (flashlight) icon, then, from the search results window you can select one item by highlighting that item and pressing the **Select** icon on the toolbar.
6. You will get a confirmation message box saying that the reference of the selected item has been sent. Now, close the EIP and Content Manager interface. Repeat the steps outlined above to select a reference for the Image item.
7. You can now save the DCR by giving a name and clicking on the **Save** button. The DCR is saved under the given name in the `data` subdirectory.



Search Results Window with Confirmation Message Box

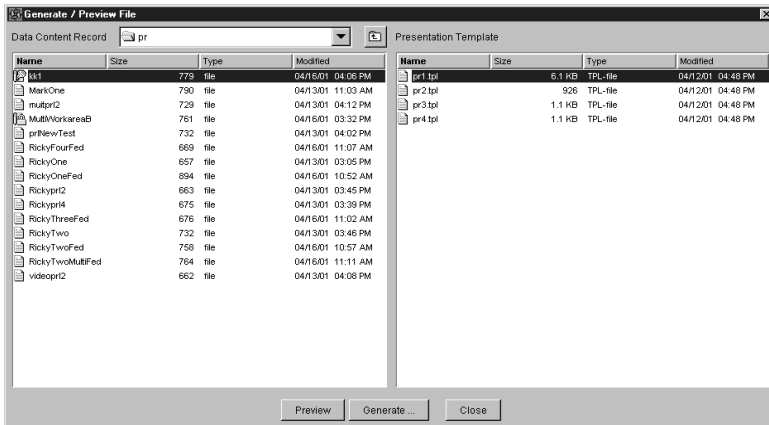
Generating or Previewing DCRs with Presentation Templates

DCRs can be previewed or associated with an output page using the *Generate Page* option from the File drop down menu in the TeamSite GUI, which brings up the Generate/Preview File window with a list of the data categories specified in the `templating.cfg` file. Turbo installation sets up one data category called `eipcmsearch` and one datatype called `pr`. Double-click on the data category name `eipcmsearch`. Then select `pr` from the next Generate/Preview File Window that opens.

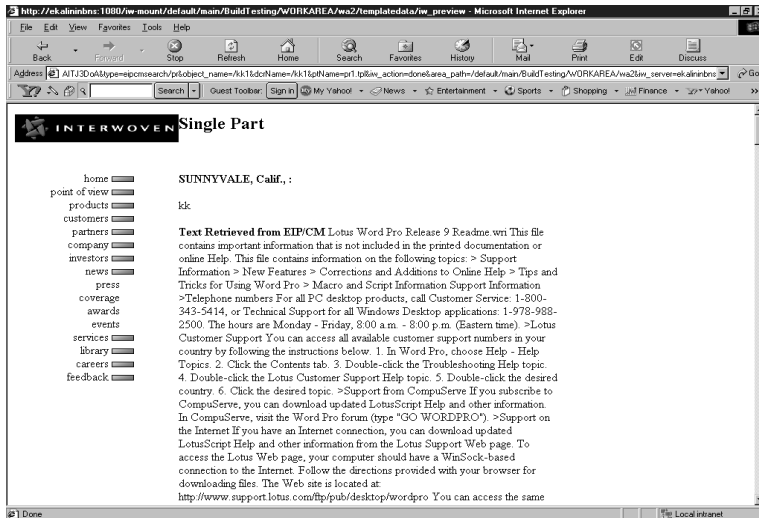


Generate/Preview File Window

Next, the DCR and Presentation Template Selection Window will open and display the DCRs under this data type. Select the DCR you want to preview from the left pane and select the PT you want the DCR to be previewed against and then click the **Preview** button. This will bring up a new browser with the contents of your DCR and the item selected from EIP and Content Manager displayed according to the PT you selected.



DCR and Presentation Template Selection Window



Preview Window

A sample preview will be displayed.



Note: The sample data that comes with EIP and Content Manager has four different data types—text, HTML, GIF and MPEG. These could be single-part content and multi part content. The type of data selected is important while previewing them using appropriate sample presentation templates. For example, `pr1` can be used to preview only single-part text documents whereas `pr2` can be used to preview any type of document. `pr3` and `pr4` demonstrate the ability to generate JSP pages; `pr3` demonstrates a federated search and `pr4` demonstrates a Content Manager direct search. `pr3` and `pr4` can only support single-part text documents.

Customization

Customizing DCTs

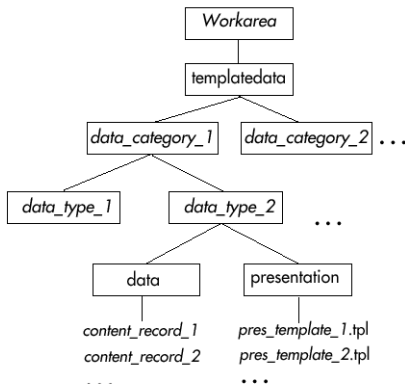
Data Capture

When users enter and save data through TeamSite Templating using a data capture template (DCT), the data is stored as XML and later used to fill in presentation templates to generate multiple renderings of the content. Users can enter data in the TeamSite Templating GUI by selecting and filling in a form generated from a DCT. When the captured data is saved, it is stored as a data content record (DCR) in the TeamSite file system.

Compare the form with the XML file containing the DCT file above. Each item tag in the DCT corresponds to a field in the form; for example, the first item tag Publish Date is the label for the first field in the form. The other items under each tag define the description for the field, database data type, type of the field, and the validation regular expression.

TeamSite Templating organizes data based on category and type. Data category is at the top level of data organizing hierarchy. Each data category contains one or more specific data types. For example, data category beverages might contain data types coffee, tea and so on.

The data storage hierarchy is shown in the following diagram:



Data Storage Hierarchy

TeamSite Data Structure Hierarchy

A sample template is shipped with Interwoven Turbo 2.0 for EIP and Content Manager. This sample contains a data type, `Press Release (pr)` under the category `eipcmsearch`. You can see the data capture template associated with this data type in the file named `datacapture.cfg` in the directory `templatedata\eipcmsearch\pr`. The `datacapture.cfg` file that is imported by the Turbo Installation into the workarea looks like:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE datacapture SYSTEM "datacapture4.5.dtd">

<data-capture-requirements type="content" name="pr">
  <!-- data-capture-requirements elements contain area elements -->

  <ruleset name="Press Release">
    <description>
      Enter Press Release information.
    </description>

    <!-- file elements contain item elements -->

    <item name="Publish Date">
      <description>date format is YYYY-MM-DD</description>
      <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="Headline">
      <database data-type="VARCHAR(100)" />
      <text required="t" maxlength="100" />
    </item>

    <item name="Secondary Headline">
      <database searchable="f" data-type="VARCHAR(200)" />
      <text maxlength="200" />
    </item>

    <item name="Introductory Paragraph">
      <database deploy-column="f" />
      <textarea rtf="t">
```

```

</textarea>
</item>

<item name="Story">
  <database deploy-column="f" />
  <replicant max="4">
    <item name="Subheading">
      <text maxlength="100" />
    </item>
    <item name="Section Paragraphs">
      <replicant max="4">
        <item name="Paragraphs">
          <textarea required="t"/>
        </item>
      </replicant>
    </item>
  </replicant>
</item>

<item name="Author">
  <database data-type="VARCHAR(40)" />
  <text maxlength="40" />
</item>

<item name="EMail">
  <database data-type="VARCHAR(60)" />
  <text maxlength="60" />
</item>
<item name="TextItem">
<description>
  Hit Search to invoke the EIP Search Panel. Then select
  a text-item from the Search Results
</description>

<text>
  <callout type='java-class' label='Search!'
  location='http://localhost/iw/java-callout-api/examples/'
  class='com.interwoven.dc100.api.EIPCallout' />
</text>
</item>

```



```
<item name="ImageItem">
<description>
    Hit Search to invoke the EIP Search Panel. Then select an
    image-item from the Search Results
</description>

<text>
    <callout type='java-class' label='Search!'
    location='http://localhost/iw/java-callout-api/examples/'
    class='com.interwoven.dc100.api.EIPCallout' />
</text>
</item>

<item name="Language">
    <database data-type="VARCHAR(10)" />
    <radio>
        <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" />
        <option label="Spanish" value="Spanish" />
        <option label="Italian" value="Italian" />
    </radio>
</item>

</ruleset>
</data-capture-requirements>
```

For more information on using TeamSite Templating, see the *TeamSite Templating User's Guide*.

Search Application

From within the TeamSite Templating data capture template, click the **Search** button to open the Enterprise Portal Client Search window. In the Enterprise Portal Client search window, users enter the criteria for their search and click on the flashlight icon to start searching content that matches their criteria. The search results are displayed in the Search Results window.

Enterprise Portal Client Application

Claim Author: NEO Subject: EQUAL Content Type: EQUAL Date: EQUAL Language: EQUAL

DLSAMPLE

IWSAMPLE

NOINDEX

Search Results

Author	Subject	ContentType	Date	Language
J. E. E. Baglin	Thin film bonding using ion b...	gif Image	2001-01-02	English
R. Evernden	The Information FrameWork	gif Image	1994-11-09	English
J. R. M. Hosking	The four-parameter kappa di...	gif Image	1995-02-22	English
J. R. M. Hosking	The four-parameter kappa di...	gif Image	1995-02-22	English
S. C. McCready	ReadMe of IBM JDK Install	html Document	2000-03-10	EnglishTraditional Chinese
M. B. Present	ReadMe of IBM JDK Install	html Document	2000-03-10	EnglishKorean
B. McNutt	Background data movement l...	gif Image	1994-10-09	English
R. A. Nobakht	A unified table-based Viterbi ...	gif Image	1996-11-15	English
C. M. Saracco	Lotus Word Pro Release 9 R...	Text Document	1998-05-11	English
R. Evernden	The Information FrameWork	gif Image	1994-11-09	English
B. L. Bradford	The Fast Staggered Transfor...	gif Image	1999-01-02	English
J. H. Hedges	OD PTF2.2.1.3 Readme	Text Document	1999-10-28	English
M. B. Present	ReadMe of IBM JDK Install	html Document	2000-03-10	EnglishKorean
C. H. Anderson	ReadMe of IBM JDK Install	html Document	2000-03-10	English
C. M. Saracco	Lotus Word Pro Release 9 R...	Text Document	1998-05-11	English
M. S. Cox	ReadMe of IBM JDK Install	html Document	2000-03-10	EnglishFrench
U. Durig	Atomic-scale metal adhesion...	gif Image	1993-06-08	English
D. J. Anchor	ReadMe of IBM JDK Install	html Document	2000-03-10	English
T. Ohta	Use of multiple representatio...	gif Image	1994-07-02	English
M. D. Thouless	Fracture mechanics for thin-fil...	gif Image	1994-10-09	English
R. T. Gouide	ReadMe of IBM JDK Install	html Document	2000-03-10	EnglishPortuguese
F. J. Taylor	ReadMe of IBM JDK Install	html Document	2000-03-10	EnglishDanish
D. Coppersmith	The Data Encryption Standar...	gif Image	1994-10-09	English
IBM	IBM Software Development Fa...	Video Clip	2001-02-30	English
A. B. Hodel	ReadMe of IBM JDK Install	html Document	2000-03-10	EnglishChinese
T. Ohta	Use of multiple representatio...	gif Image	1994-07-02	English
F. Gozzo	Recursive least-squares seq...	gif Image	1994-12-02	English
P. F. Freeman	OD PTF2.2.1.6 Readme	Text Document	2000-04-09	English
Charles J. Bontempo	IBM eNetwork Personal Com...	Text Document	1998-06-18	English

198 Items found. Search completed.

Search Results Window

Search Results

When the search results are returned, highlight an item, then click the **Select** button on the toolbar to bring a reference of the selected item to the templating environment and save it as part of the DCR.

The key items retrieving content from EIP and Content Manager are the last two fields in this DCT, `Text Item` and `Image Item`. These fields make use of the Java call-out facility in templating to launch an EIP search GUI that displays an EIP search panel to search in Content Manager. After running the search, selecting a particular item of interest places a reference to that content item to the DCR. The information regarding the callout, such as the type of callout, location, and such, is specified in the callout tag. Use the `datacapture.cfg` file from the Press Release sample when creating your own templates. Depending on your requirements the whole DCR form can be customized including the number of callout buttons required.

For more details on how to create `datacapture.cfg`, see the *TeamSite Templating Classic Developer's Guide*.

Customizing Presentation Templates

In addition to customizing the data capture, the presentation template can be customized as well.

After data is captured and stored as a data content record (DCR), you can combine the DCR with a presentation template. The result is a generated output file that displays the content in a way defined by the presentation template. The same DCR can be used to generate multiple output files using different presentation templates. The resulting pages contain the same content but displayed in different formats according to the associated presentation templates.

The Press Release sample shipped with the Interwoven Turbo for EIP and Content Manager contains four presentation templates: `pr1`, `pr2`, `pr3`, and `pr4`. These templates demonstrate the capabilities of the presentation templating mechanism. The sample templates are available in the directory `templatedata\eipcmsearch\pr\presentation`.

Presentation Template 1 (Pr1)

`pr1` is listed below. PTs are XML files that can contain HTML, Interwoven tags, and Perl code.

```
<?xml version="1.0" encoding="UTF-8"?>
<iw_pt name="PressRelease"><![CDATA[
<HTML>
  <!-- Begin CDATA Tag -->
  <!-- HTML stuff is enclosed in CDATA tag -->

  <HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=UTF-8">
  </HEAD>
  <body bgcolor="#FFFFFF" link="#0033CC" vlink="#0033CC" alink="0000FF"
    TEXT=#000000 BACKGROUND="/templatedata/eipcmsearch/pr/images/
pixel.gif">
  <TABLE WIDTH="720" VALIGN="top" CELLPADDING="0" CELLSPACING="0"
  BORDER="0">
    <TR><TD WIDTH="200" VALIGN="TOP">
```

```
<IMG SRC="/templatedata/eipcmsearch/pr/images/iw-logo-small.gif"
WIDTH="220" HEIGHT="40" BORDER="0"/>
```

```
]]> <!-- End of CDATA Tag -->
```

```
<P>
```

```
<iw_include file='templatedata/eipcmsearch/pr/iwprnavbar.html' />
<BR></BR> </P>
```

```
<![CDATA[ <!-- Begin CDATA Tag -->
</TD><TD VALIGN="top" WIDTH="510">
```

```
]]> <!-- End of CDATA Tag -->
```

```
<iw_perl>
```

```
<![CDATA[
```

```
sub get_image_id
```

```
{
```

```
use URI::Escape;
```

```
my $itemID = iwpt_dcr_value('dcr.ImageItem');
```

```
return (uri_escape($itemID));
```

```
}
```

```
sub get_text_id
```

```
{
```

```
use URI::Escape;
```

```
my $itemID = iwpt_dcr_value('dcr.TextItem');
```

```
return (uri_escape($itemID));
```

```
}
```

```
sub get_was_server
```

```
{
```

```
use TeamSite::Config;
```

```
$iw_home = TeamSite::Config::iwgethome();
```

```
my $prop= "$iw_home\\local\\config\\iweip.properties";
```

```
open(PROP, "$prop") or die "Could'nt open $prop \n";
```

```
while(<PROP>)
```

```
{
```

```
my $line = $_;
```

```
if ($line =~ /SERVER=(.*)/){ $wasserver = $1;}
```

```
}
```

```
return ($wasserver);
```

```
}
```



```
sub get_was_port
{
use TeamSite::Config;
$iw_home = TeamSite::Config::iwgethome();
my $prop= "$iw_home\\local\\config\\iweip.properties";
open(PROP, "$prop") or die "Could'nt open $prop \n";
while(<PROP>)
{
my $line = $_;
if ($line =~ /PORT=(.*)/){ $wasport = $1;}
}
return ($wasport);
}

sub get_web_app
{
use TeamSite::Config;
$iw_home = TeamSite::Config::iwgethome();
my $prop= "$iw_home\\local\\config\\iweip.properties";
open(PROP, "$prop") or die "Could'nt open $prop \n";
while(<PROP>)
{
my $line = $_;
if ($line =~ /WEBAPP=(.*)/){ $webapp = $1;}
}
return ($webapp);
}

sub call_servlet
{
    use HTTP::Request::Common qw(POST);
    use LWP::UserAgent;
    my $itemid = $_[0];
    my $wasserver = get_was_server();
    my $wasport = get_was_port();
    my $webapp = get_web_app();
    my $url = "http://" . $wasserver . ":" . $wasport . "/" . $webapp . "/" .
RetrieveDoc" ;
    my $ua = LWP::UserAgent->new();

    my $req = POST($url, [ itemid => $itemid ] );
```

```

my $for_debug = $req->as_string();
my $res = $ua->request($req);

if ($res->is_success()) {
    my $resulting_doc = $res->content();
return $resulting_doc;
}
else
{
return "\nHTTP Request Failed.\n";
}
}

```

```
]]>
```

```
</iw_perl>
```

```

<!-- Begin content area -->

<!-- Headline -->
<H2> <iw_value name='dcr.Headline' /></H2>

<P></P>
<br></br>

<!-- Secondary Headline -->
<h3> <iw_value name='dcr.Secondary Headline' /> </h3>

<!-- Date -->
<P> <B>SUNNYVALE, Calif., <iw_value name='dcr.Date' />:</B></P>

<!-- Introductory Paragraph -->
<P><iw_value name='dcr.Introductory Paragraph' /> </P>

```



```
<!-- Story -->
<iw_iterate list='dcr.Story' var='story'>

  <!-- Subheading -->
  <iw_value name='story.Subheading' />

  <!-- Paragraphs -->
  <iw_iterate list='story.Section Paragraphs' var='para_value'>
    <p><p><iw_value name='para_value.Paragraphs' /></p></p>
  </iw_iterate>

  </iw_iterate>
<p><B> Text Retrieved from EIP/CM </B>
<iw_perl>
  my $item = iwpt_dcr_value('dcr.TextItem');
  my $content = call_servlet($item);
  iwpt_output($content);
</iw_perl>

</p>

<!-- Insert 'aboutIW.html' file -->
<p>
<iw_include file='templatedata/eipcmsearch/pr/aboutIW.html' />
</p>

<p>For more information on the company and
its software solutions, visit the Interwoven Web site at
<a href="http://www.interwoven.com">www.interwoven.com</a>
or email <a href="mailto:{iw_value name='dcr.EMail'}">
  <iw_value name='dcr.EMail' /></a>

</p>

<!-- HTML stuff is enclosed in CDATA tag -->
<![CDATA[  <!-- Begin CDATA tag -->

  <p>
  <TABLE WIDTH=520 BORDER=0 CELLSPACING=10 CELLPADDING=0>
  <TR>
    <TD COLSPAN=2 BGCOLOR=#999999>
```

```

        <IMG SRC="/templatedata/eipcmsearch/pr/images/pixel.gif">
    </TD>
</TR>
<TR>
    <TD COLSPAN=2>    <BR>
    <!-- Begin question -->
    <CENTER>
    <FONT SIZE=3><B>
    <A HREF="/customers/profiles/bestbuy.html">How is Best Buy pushing
        billions of dollars in business towards the Web?</A>
    </B></FONT>
    </CENTER>

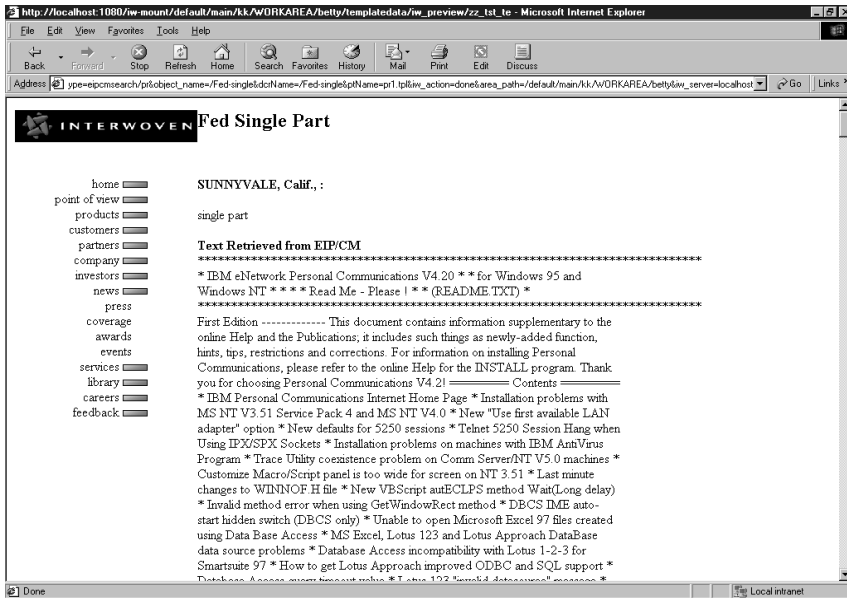
    <!-- End question -->
</TD>
</TR>
<TR>
    <TD COLSPAN=2>&nbsp;
    </TD>
</TR>
<TR>
    <TD COLSPAN=2 BGCOLOR=#999999>
    <IMG SRC="/templatedata/eipcmsearch/pr/images/pixel.gif">
    </TD>
</TR>
<TR>
    <TD WIDTH=250 VALIGN="top"><IMG SRC="/templatedata/eipcmsearch/pr/
images/pixel.gif" WIDTH=250 HEIGHT=1>
    <FONT SIZE="1" COLOR=#666666> <B>Global Headquarters</B><BR>
    Interwoven, Inc.<BR>    1195 W. Fremont Ave. #2000<BR>
    Sunnyvale, CA 94087 US<BR>    Phone: (408) 774-2000<BR>
    </FONT>
</TD>
    <TD WIDTH="250" VALIGN="top"><BR><FONT SIZE=1 COLOR=#666666>
    <B>How Can We Serve You?</B><BR>
    Let us know at:<A HREF="mailto:info@interwoven.com">
    info@interwoven.com</A><BR>
    or <A HREF="/cgi-bin/reg">Register</A> and we will contact
you!<BR>
    <BR><NOBR>Web Team: <A HREF="mailto:webteam@interwoven.com">
    webteam@interwoven.com</A>

```

```
</NOBR></FONT>
</TD>
</TR>
<TR>
  <TD COLSPAN=2 ALIGN=CENTER>    <BR>
  <FONT COLOR=#666666>    <P CLASS="copyright">
  <A HREF="/copyright.html">    Copyright &copy; 1999</A>
  Interwoven, Inc. All rights reserved.    </P>    </FONT>
  </TD>
</TR>
</TABLE>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
]]>    <!-- End CDATA tag -->

</iw_pt>
```

The Perl functions included in this presentation template are used to retrieve the text document content by item ID and embed the actual content into an HTML file during page generation process. The `iweip.properties` file stored in `iw-home\local\config` directory contains information about the WebSphere Application Server, the port, and the Web application that will be used by the Perl functions to issue `HTTP POST` requests to contact the `RetrieveDoc` servlet that is registered with a specified Web application within WebSphere Application Server. The Perl functions `get_was_server()`, `get_was_port()`, and `get_web_app()` within the presentation template return the server name, server port, and Web application name respectively from the above property files. The function `call_servlet()` uses the functions mentioned previously to construct a URL and issue an `HTTP POST` request to the `RetrieveDoc` servlet. In this example, only a single-part text document that contains plain text can be used with this presentation template to generate a static HTML embedding the actual text content. The `RetrieveDoc` servlet retrieves the content by the item ID through EIP API functions. The sample output using the presentation template `perl` is shown below:



Preview Window Generated by Presentation Template 1 (pr1)

Presentation Template 2 (Pr2)

The following presentation template is the pr2 sample.

```
<?xml version="1.0" encoding="UTF-8"?>
<iw_pt name = "PressRelease3">

<!-- HTML stuff is enclosed in CDATA tag -->
<![CDATA[<!-- Begin CDATA tag -->
<HTML>
<head>
<title>Test Image Retrieve</title>
<body>
]]>
<iw_perl>
<![CDATA[
sub get_image_id
{
    use URI::Escape;
    my $itemID = iwpt_dcr_value('dcr.ImageItem');
```

```
    return (uri_escape($itemID));
}
sub get_text_id
{
    use URI::Escape;
    my $itemID = iwpt_dcr_value('dcr.TextItem');
    return (uri_escape($itemID));
}

]]>

</iw_perl>

    <p><b> This is the Image Retrieved from CM Direct </b></p>
    <![CDATA[<IMG SRC="/
RetrieveDoc?itemid=]]><iw_perl>iwpt_output(get_image_id());</
iw_perl><![CDATA[ "/>]]>

    <p><b> This is the Text Content Retrieved from CM Direct </b></p>
    <![CDATA[<a href="/
RetrieveDoc?itemid=]]><iw_perl>iwpt_output(get_text_id());</
iw_perl><![CDATA[ ">Click here</a>

</body>
</HTML>
]]>

</iw_pt>
```

This presentation template illustrates the use of HTML `` and `<A HREF>` tags in conjunction with the `RetrieveDoc` servlet to retrieve the text item and image item, as an alternative method for retrieving the content. This presentation template can support multi-part document items as well as single-part document items.

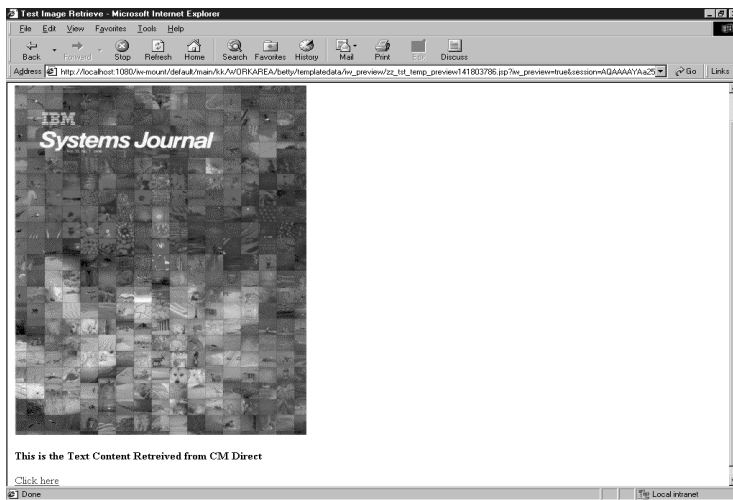
This presentation template, unlike `pr1`, generates a JSP page with an `` reference to the image content and an `<A HREF ...>` URL reference to the other content. These links can be a

reference to multi-part content as well. The multi-part content can be displayed by the `viewframes.jsp` file, which is used by the `RetrieveDoc` servlet.

The `viewframes.jsp` file, other associated JSP files, and supporting files are copied into the `iweip` directory of the workarea specified during Turbo Installation. Users can customize the `viewframes.jsp` file and any related JSP file provided that they do not change the file name and the structure of the `iweip` directory in the workarea. For more information on how to customize the servlet and JSPs to handle multi-part documents, refer to the “Customizing the RetrieveDoc Servlet” section at the end of this chapter.

The servlet classpath of the Web application that contains the `RetrieveDoc` servlet should have all the EIP-required JAR files: for example, `cmb71.jar`, `cmbSDK71.jar` and `cmbview71.jar`. Usually, these three JAR files are available in the root installation of EIP, for example, in `c:\cmbroot\lib`.

When this presentation template displays an image it appears like the following:



Preview Window Generated by Presentation Template 2 (pr2)

Presentation Template 3 (Pr3)

Presentation template `pr3` illustrates the use of JSP with EIP API functions to perform federated searching in retrieving the content. This sample PT demonstrates that it is possible to generate a complete JSP file. The code implements direct EIP Java API calls to retrieve the content rather than using the servlet and the output is written to the output stream as String. The JSP generated by this sample presentation template can handle only single-part text contents. The JSP can be modified to handle other types of contents. The template is listed below:

```
<?xml version="1.0" encoding="UTF-8"?>
<iw_pt name = "PressRelease3">

<!-- HTML stuff is enclosed in CDATA tag -->
<![CDATA[<!-- Begin CDATA tag -->
<HTML>
<P> <b> Text Retrieved from Fed using APIs </b> </p>
<p>
<%@ page
    language="java"
    import="java.util.*, com.ibm.mm.beans.*, java.io.*"

%>
<%
    CMBCConnection connBean = null;
    CMBCConnectionPool cmPoolBean = null;
    String itemid = "]]><iw_value name='dcr.TextItem' /><![CDATA[";
    CMBItem item = new CMBItem(itemid);

    cmPoolBean = new CMBCConnectionPool();
    cmPoolBean.setDsType("Fed");
    cmPoolBean.setMaxFreeConnections(100);
    cmPoolBean.setMaxFreeConnectionsPerUserid(100);

    connBean = cmPoolBean.getConnection("cmbdb", "cmbadmin", "password");
    CMBDataManagement dataBean = connBean.getDataManagement();
    dataBean.setDataObject(item);
    dataBean.retrieveItem();
    CMBObject content = dataBean.getContent(0);
    String mimeType = content.getMimeType();
    response.setContentType(mimeType);

%>
```

```

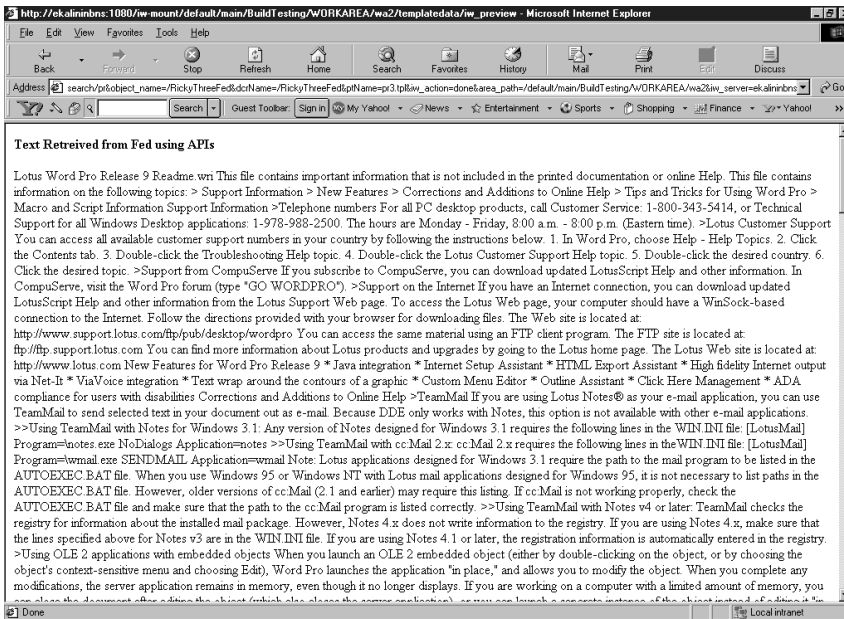
</p>
  <p>  <% out.write(new String(content.getData()));%> </p>
</HTML>

]]>

</iw_pt>

```

Using `pr3` to display text content would look like the following:



Preview Window Generated by Presentation Template 3 (pr3)

Presentation Template 4 (Pr4)

Presentation template `pr4` illustrates the use of JSP with EIP API functions to perform Content Manager direct searching in retrieving the content. This sample PT demonstrates that it is possible to generate a complete JSP file. The code implements direct EIP Java API calls to retrieve the content rather than using the servlet, and the output is written to the output stream as String. The JSP generated by this sample presentation template can handle only single-part text contents. The JSP can be modified to handle other types of contents. The template is listed below:



```
<?xml version="1.0" encoding="UTF-8"?>
<iw_pt name = "PressRelease3">

<!-- HTML stuff is enclosed in CDATA tag -->
<![CDATA[<!-- Begin CDATA tag -->
<HTML>
<P> <b> Text Retrieved from CM using APIs </b> </p>
<p>
<%@ page
    language="java"
    import="java.util.*, com.ibm.mm.beans.*, java.io.*"

%>
<%
    CMBConnection connBean = null;
    CMBConnectionPool cmPoolBean = null;
    String itemid = "]]<iw_value name='dcr.TextItem'/><![CDATA[";
    CMBItem item = new CMBItem(itemid);

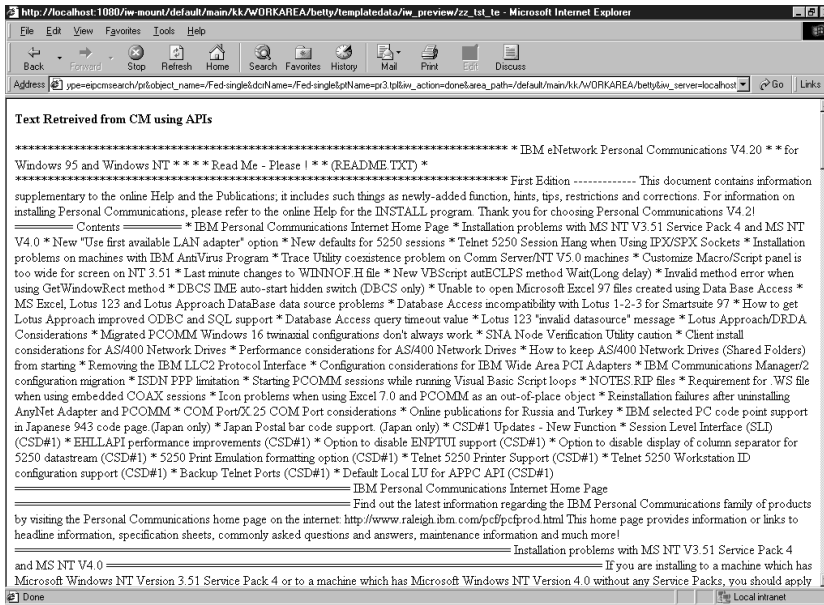
    cmPoolBean = new CMBConnectionPool();
    cmPoolBean.setDsType("DL");
    cmPoolBean.setMaxFreeConnections(100);
    cmPoolBean.setMaxFreeConnectionsPerUserid(100);

    connBean = cmPoolBean.getConnection("libsrvrn","frnadmin","password");
    CMBDataManagement dataBean = connBean.getDataManagement();
    dataBean.setDataObject(item);
    dataBean.retrieveItem();
    CMBObject content = dataBean.getContent(0);
    String mimeType = content.getMimeType();
    response.setContentType(mimeType);
%>
</p>
    <p> <% out.write(new String(content.getData()));%> </p>
</HTML>

]]>

</iw_pt>
```

Using pr4 to display text content would look like the following:



Preview Window Generated by Presentation Template 4 (pr4)

Customizing the RetrieveDoc Servlet

About the RetrieveDoc Servlet

Two of the sample presentation templates use the `RetrieveDoc` servlet to retrieve content from EIP or Content Manager. This servlet requires an item ID (unique content reference ID) plus other information that is stored in a properties file called `iveip.properties`. The location of this properties file is specified as an initialization parameter called `propFilePath` for the servlet. The default value for this parameter is set to the root of the directory for EIP Server installation on the WebSphere Application Server computer (for example, `C:\cmbroot\lib`). The properties file contains the following information.

- Root directory—Where EIP Server is installed – `C:\cmbroot`
- Search type—`Fed` or `CM` (for federated and Content Manager Direct searches, respectively. Note: spelling is case-sensitive)
- Server name—Use `database` if `Fed` is used or `library server name` if `CM` is used)
- User ID—For a federated search, this is the EIP user ID; for searching Content Manager directly, this is the Content Manager user ID
- Password—For the user ID specified

The properties file is read during the servlet loading when the Web application starts. If the user needs to change any value in `iveip.properties`, stop and restart the Web application that contains this servlet to reinitialize the servlet after editing the properties file.

Note: A local copy of the `iveip.properties` file is also required on the templating client computer. The location of the property file is in the `Templating client directory\eipcm` directory. This file contains all the above parameters plus some other init parameters for the search UI. The values for all the above parameters need to match with the values specified in the `iveip.properties` file on the WebSphere Application Server computer, except that the search type for Content Manager direct is `CM` rather than `DL`.

RetrieveDoc Usage Scenarios

Scenario 1

Only presentation template `pr1` can be used in this scenario. The `RetrieveDoc` servlet is registered with the Web application called `iweip`, which is configured and created during the Turbo installation. The servlet classpath for the `RetrieveDoc` servlet is set to `C:\cmbroot\lib\iweip.jar` and other required JAR files for EIP toolkits, where `cmbroot` is the directory for EIP Server installation. For more information on the classpath and required JAR files, refer to the *Managing Enterprise Information Portal* manual or the last section of this chapter for details. This Web application is configured and tailored to be used for `pr1` within this integration, therefore changes are not allowed. Thus, in the sample, the full servlet URI is:

```
http://<websphere server>:<port>/iweip/RetrieveDoc?itemid=
```

This makes a hardcode reference to the `iweip` Web application and this is intended to be used with a PT that only generates static HTML pages with embedded single-part plain text documents.

Scenario 2

Presentation template `pr2` generates JSP pages that use the `RetrieveDoc` servlet registered within the sandbox Web application configured with the workarea. These JSP pages are served up by the application server at runtime. In `pr2`, the image is retrieved by invoking the `RetrieveDoc` servlet using a relative reference. The servlet in this scenarion is registered in the Web application (created by heavy weight sandbox allocation) that connects to the TeamSite workarea. The servlet classpath is set up during heavyweight sandbox allocation.

For example:

The image will be downloaded by the browser:

```
<IMG SRC="/RetrieveDoc?itemid=<itemid>">
```

The text item is retrieved by clicking on the link that refers to:

```
<A HREF="/RetrieveDoc?itemid=<itemid>">link text</A>
```

If the item ID for the above text item refers to a single-part document, the document is served immediately. If the item ID refers to a multi-part document, then the `RetrieveDoc` servlet will redirect the request to the `viewframes.jsp` file to display the multi-part document in a frameset. This JSP and all the other supporting files are available in the `iweip` directory of the workarea specified during Turbo Installation. This scenario that uses the `pr2` sample PT assumes the use of the `RetrieveDoc` servlet that comes with the Turbo product and therefore assumes the use of the `viewframe.jsp` file as well. User can choose to customize `viewframes.jsp` as needed or replace it with your own JSP. The name for the customized or replacement JSP for `viewframes.jsp` must be the same, and it must be placed in the `iweip` directory of the workarea. In summary, users are encouraged to customize the frameset JSPs but the `iweip` directory structure and naming convention of the `viewframes.jsp` cannot be changed.

Scenario 3

If users choose not to use the `RetrieveDoc` servlet shipped with the Turbo, they are encouraged to develop their own equivalent of this servlet and register it with the Web application that is configured with their workarea during heavyweight sandbox allocation (must define the new servlet in the `config.xml` file). The source code of `RetrieveDoc` servlet is shipped with the Turbo installation and can be found in the `install/websphere/example/RetrieveDoc_Example` directory. Users are welcome to refer to this reference implementation. In this scenario, users are no longer required to use `viewframes.jsp`, or any supporting files that reside in the `iweip` directory, to support multi-part document display.

Production Deployment Using RetrieveDoc

Using the `RetrieveDoc` servlet in production that shipped with Turbo product:

- Ensure that the `RetrieveDoc` servlet (or equivalent) is registered with your production Web application.
- Ensure that the servlet classpath is set to point to the JAR file containing the servlet or other JAR files required by EIP APIs. The servlet is packaged in a JAR file called `iweip.jar` that comes with the Turbo product. User can find a copy of `iweip.jar` residing in the `cmbroot\lib` directory on the WebSphere Application Server computer (in the development environment where the Turbo product is installed) where `cmbroot` is the EIP server root directory, because this JAR file will be copied into the above location by the Turbo installation.

Here is a complete list of all required JAR files by RetrieveDoc servlet:

- `cmbroot\lib\iweip.jar`
- `cmbroot\lib\cmb71.jar`
- `cmbroot\lib\cmbview71.jar`
- `cmbroot\lib\cmbsdk71.jar`
- `cmbroot\lib\ics.jar`
- `cmbroot\lib\xerces.jar`
- `cmbroot\lib\lotuskms.jar`
- `cmbroot\lib\essrv.jar`
- `cmbroot\lib\esclisrv.jar`
- `cmbroot\java\db2java.zip`

Note: The Turbo installation will only set up the first four JAR files in the servlet classpath because the Turbo reference implementation will use only the Content Manager repository. If the EIP is set up to use a repository other than Content Manager, make sure the servlet classpath includes the JAR files listed above.

- Ensure that initialization parameter `propFilePath` for the servlet is set up to point to the `iweip.properties` file in the production environment.
- Ensure that the `iweip` directory that contains all related assets for the frameset reside in the top of the document root.
- Deployment from the TeamSite staging area or edition to a production environment (including JSPs, HTML, images, servlets, DCRs, and so on) can be accomplished by a combination of tools using TeamSite OpenDeploy, DataDeploy, and WebSphere publishing tools (mainly registering servlet and EJBs). Refer to the *OpenDeploy Administration Guide* and *DataDeploy Administration Guide* manuals for details on how to perform deployment from a TeamSite environment to a production environment.

Chapter 5

Troubleshooting

This chapter identifies some potential problems you might encounter when installing the Turbo product and when working with the Turbo components, TeamSite, EIP and Content manager. For each potential error, some general troubleshooting techniques are proposed.

Potential problems (and solutions) include the following:

- If you are able to preview and retrieve content during searching within the templating Java UI, but the content is not being displayed during previewing or after page generation of JSP or HTML.

Check that the `iweip.properties` files on the WebSphere Application Server and the Templating Client are set to operate in the same way. The two properties files must be set to access Content Manager directly or to access EIP; they must be set to the same search type and DB access information.

- EIP/Content Manager client installation failed.

Templating client program might be running. Close the templating client window.

- Logon failed to server `libsrvn` when performing federated searching.

Sample error message:

```
com.ibm.mm.beans.CMBException: Unable to connect to the server.:  
java.lang.ClassNotFoundException: COM.ibm.db2.jdbc.app.DB2Driver.
```

Add `db2installdir\java\db2java.zip` at the end of the `lax.class.path` of the `tst.lax` file in the directory where you installed the Templating Client; the default is

```
c:\Program Files\Interwoven\TemplatingClient.
```

- Click on the DCT search button and nothing happens.

This indicates an installation problem for the Turbo component on the templating client; or may also indicate faulty configuration. Unfortunately there won't be any error messages displayed prompting a return to Templating Java UI. This might require the user to uninstall and re-install the Turbo component.

- Error 404 while previewing or generating pages from TeamSite.

Make sure you have allocated a heavyweight sandbox to the workarea that you are operating in by using the special menu item called SandboxPool in the TeamSite Menu bar. Ensure that only heavyweight sandbox can be used with this version of the integration.

- Error 500 while previewing or generating page from TeamSite.

Check for a mismatched search type on the templating client computer and WebSphere application server computer. For example, `CM direct` on the client and `Fed` on the server.

Another possibility is the servlet classpath might not be set up correctly; refer to the last section of Chapter 4 for details on the required servlet classpath.

Content Manager Sample Loader Program

The sample loader program, and the files associated with it, are available in the `cmsample` subdirectory of the `iweipclient` installation directory. Files and directories that are required for this loading program are:

- `iwsample`—directory containing the actual data files such as the text, GIF and MPEG files.
- `index.exe`—the application that creates the index
- `IWSAMPLE.dat`—input file to the loader program
- `IWSAMPLE.log`—log file for the loader program
- `IWSampleLoader.class`—Java loader program

Ensure that the JDK is included in the environment variable `PATH` by setting it to include the Java directory.

Type the following in the command window:

```
Java -classpath <current directory>;<eip root
directory>\lib\cmb71.jar;<eip root directory>\lib\cmbSDK71.jar
IWSampleloader <library server> <object server> <userid> <password>
```

Where:

- `eip root directory` is the directory where EIP is installed
- `library server` is the name of the library server component
- `object server` is the name of the object server component
- `userid` is the valid user ID in Content Manager
- `password` is the password for the user ID mentioned above

A sample command might look like the one below assuming that your current directory is the one that contains the loader program and related files:

```
Java -classpath .;c:\cmbroot\lib\cmb71.jar;c:\cmbroot\lib\cmbSDK71.jar  
IWSampleLoader LIBSRVRN OBJSRVRN frnroot password
```

This will start the loading program by creating the `IWSAMPLE` index, if it does not exist, and loading the sample data available in the `iwsample` directory.

A known issue is that if the index `IWSAMPLE` does not exist, the program terminates after creating the index. Executing the program one more time loads the data correctly.

Note: If `library server`, `object server`, `userid` and `password` are omitted from the command, the default values, respectively, for these arguments are: `LIBSRVRN`, `OBJSRVRN`, `frnadmin`, and `password`.