

IMS Web User's Guide

webu-0002-01

March 10, 1998

Candace Garcia
IBM Corporation



IMS Web User's Guide

IMS Web user's guide

© Copyright International Business Machines Corporation 1997. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Trademarks	v
Chapter 1. Overview of IMS Web	1
IMS Web Development component	2
IMS Web Runtime component	3
Net.Data support	3
Chapter 2. Prerequisites	5
IMS Web Development component	5
IMS Web Runtime component	5
Net.Data	6
Prerequisites	6
Installation information for the IMS Web language environment	6
Chapter 3. IMS Web Development	7
How to install the IMS Web Development component	7
Starting IMS Web Studio	8
Creating a project in IMS Web Studio	8
Associating MFS source with IMS Web Studio project	9
Parsing files	11
Defining transactions	12
Defining classes	13
Defining hosts	15
Defining platforms	16
Generating classes	17
Directory structure and associated files	17
Building an executable file	19
IMS Web transactions	20
Nonconversational transactions	20
IMS conversational transactions	20
User-conversational transactions	22
IMS Web protocol for conversations	22
Notes for version 1 users	24
Java virtual machine considerations	25
Chapter 4. IMS Web Runtime component	27
How to install the IMS Web Runtime component	27
Incorporating project executable files into your Web server environment	29
For IMS Web only	29
For Net.Data	29
IMS Web Runtime commands	30
HWSCMD TRACE command	30
HWSCMD DISPLAY CONV command	31
HWSCMD EXIT command	33
IMS Web Runtime data conversion	34
Web Server Platform NT, OS/2, AIX, or Solaris	34
Web Server Platform OS/390	34
Considerations when using ICS for OS/390	34

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX	IBM
IMS	Net.Data
OpenEdition	OS/2
OS/390	RACF
VisualAge	

Microsoft and Windows NT are registered trademarks of Microsoft Corporation.

Java, JDK, and JRE are trademarks of Sun Microsystems, Inc.

Other company, product, and service names may be trademarks or service marks of others.

Chapter 1. Overview of IMS Web

The following topics are described in this section:

- “IMS Web Development component” on page 2
- “IMS Web Runtime component” on page 3
- “Net.Data support” on page 3

IMS Web gives you a way to reuse your existing IMS transactions in the World Wide Web environment by providing a complete end-to-end solution, including runtime libraries and a tool that generates all the code required to build a Web-based application that accesses existing IMS transactions.

IMS Web consists of the following components:

- IMS Web Development

IMS Web Development includes:

- IMS Web Studio: a GUI tool that allows you to:
 - Select and download MFS source files for IMS transactions
 - Generate both a CGI program and HTML file for IMS Web-driven applications from the MFS source files
 - Generate source for a Net.Data transaction DLL and macro for Net.Data-driven applications from the MFS source files
- A set of library and header files that are required to build the generated code into an executable that runs in the IMS Web Runtime environment

IMS Web Development supports the following platforms:

- Windows NT and Windows 95
- OS/2
- AIX
- Sun Solaris
- OS/390

- IMS Web Runtime

IMS Web Runtime provides a set of functions that are called by the IMS Web Studio generated code.

The IMS Web Runtime component provided functions include:

- Communication to IMS TCP/IP OTMA Connection via TCP/IP
- MFS-like message formatting and platform data conversion

The IMS Web Runtime component supports the following platforms:

- Windows NT and Windows 95
- OS/2
- AIX
- Sun Solaris
- OS/390

Attention: IMS Web can be used by itself or in conjunction with Net.Data. If you plan to use Net.Data to submit your IMS transactions, the IMS Web Runtime component is a prerequisite for Net.Data.

IMS Web Studio generates files in two directories under the project directory: cgi and dtw. The cgi directory contains all of the files needed for an IMS Web-driven transaction. The dtw directory contains all of the files needed for the IMS Web backend of a Net.Data-driven transaction.

The following elements are generated in the cgi subdirectory:

- C++ classes and source code for a CGI-BIN program that runs an IMS transaction from a workstation and dynamically creates an output HTML form to display the results of that transaction
- An input HTML form for submitting transaction input data and for running the transaction CGI-BIN program

The following elements are generated in the dtw subdirectory:

- C++ classes and code for a transaction DLL that runs an IMS transaction from a workstation
- Input and output HTML for the transaction packaged in a Net.Data macro

What's new in Release 210:

- Support of different transaction types:
 - Nonconversational transaction support: same as pre Release 210
 - IMS-conversational transaction support: IMS transactions that use the SPA to save state data between iterations of the conversation
 - User-conversational transaction support: IMS transactions that use a user-defined way to save state data between iterations of the conversation
- Options to generate HTML to gather host information from a browser user:
 - IMS TOC hostname
 - IMS TOC port
 - Target IMS XCF member name
- Options to generate HTML to gather user information from a browser user:
 - userid
 - groupname
 - password
- Redesigned Studio tool. The IMS Web Studio is now a TaskGuide that leads you through the process of generating IMS Web code, step by step. The tool was designed in Java so the tool can run on all platforms.

IMS Web Development component

The IMS Web Development component (see “Chapter 3. IMS Web Development” on page 7) includes:

- IMS Web Studio
- C++ header files and library files required to compile and link the generated C++ files for the IMS Web-driven CGI-BIN program or Net.Data-driven transaction DLL on the NT, OS/2, AIX, Sun, or OS/390 platform

IMS Web Studio is a Java application that runs on JDK or JRE 1.1.3 or above and provides the following functions:

- Allows you to browse and download a set of MFS source files for a transaction
- Generates, from the downloaded files, C++, HTML, and Net.Data macro files for the transaction
- Generates .mak files to compile and link the generated C++ files to create an executable CGI-BIN program and transaction DLL

You move the CGI-BIN program and input HTML form, and/or transaction DLL and Net.Data macro, to your Web server to provide access to the IMS transaction.

IMS Web Runtime component

The IMS Web Runtime component (see “Chapter 4. IMS Web Runtime component” on page 27) consists of libraries (which are installed on the Web server and are invoked by the generated CGI-BIN program or transaction DLL), conversion tables for the IMS Web supported languages, and an executable file, HWSCMD.EXE.

The conversion tables are used to convert text data in IMS messages so that it is displayed properly on all supported platforms, in all supported languages. HWSCMD.EXE is the command line executable file that is used to start and stop tracing of activity by the IMS Web Runtime component, as well as display and terminate IMS Web conversational transactions.

For a Windows NT, OS/2, or OS/390 Web server, the runtime component libraries are Dynamic Link Libraries (DLLs); for an AIX or Sun Web server, the runtime component libraries are shared libraries (.a files for AIX and .su for Sun). These libraries provide the following functions:

- Communication to and from IMS TCP/IP OTMA Connection, which accesses IMS
- Conversion of IMS message data between Web server and host platforms
- MFS-like formatting of messages

Net.Data support

Net.Data is an IBM software product that allows Web developers to build Web applications using Net.Data Web macros. Net.Data provides connectivity to a variety of data sources and applications including relational databases, flat files, registries, system services, and IMS databases through IMS transactions.

Net.Data is an application that runs on an HTTP Web server. You can use it as a CGI application or as a Web server API application. When Net.Data is used as a CGI application, it is invoked as the executable DB2WWW. When Net.Data is used as a Web server API application with NSAPI, ICAPI, or ISAPI, the server calls it as a dynamic linked library (DLL) or shared library.

Restriction: The IMS Web language environment of Net.Data is only supported when Net.Data runs as a CGI application. It is not supported with Net.Data as ICAPI.

Language environments are Net.Data's interface to data and applications. Each language environment provides a specific interface. For example, Net.Data provides

language environments to access DB2 databases, REXX, PERL, and C applications, file systems, Web registries, Java applets, and IMS transactions through IMS Web.

The Net.Data Web macro is a file that contains a series of macro language, HTML, and language environment-specific statements. When a Web server receives a URL that refers to Net.Data, the Web server starts an instance of Net.Data and passes essential information to it, including the name of the Web macro that it needs. Net.Data reads and parses through the Web macro and interprets the statements.

When a Net.Data %FUNCTION statement is encountered in the Web macro, it loads the library of the requested language environment and passes language-specific information to the language environment for processing. The language environment processes the information and returns the results to Net.Data.

After all parsing and language environment processing are completed, all that remains is HTML text that any Web browser can interpret. The HTML text is passed back to the Web server and Net.Data terminates. The resulting HTML is passed to the Web browser.

IMS Web Studio generates the Net.Data Web macro and the source code for the associated DLL or shared library used to process an IMS transaction via IMS Web. The Net.Data Web macro includes both the input and output HTML for the IMS transaction, as well as a %FUNCTION statement to invoke the IMS Web language environment, which, in turn, dynamically loads and executes the transaction DLL or shared library.

Chapter 2. Prerequisites

The following topics are described in this section:

- “IMS Web Development component”
- “IMS Web Runtime component”
- “Net.Data” on page 6

Following are the prerequisites for enabling IMS transactions on the World Wide Web using IMS Web. If you plan to use Net.Data to submit your IMS transactions, the IMS Web Runtime component is a prerequisite for “Net.Data” on page 6.

IMS Web Development component

The IMS Web Development component consists of the IMS Web Studio tool and a set of library and header files that are required to build the generated code into an executable that runs in the IMS Web Runtime environment.

The IMS Web Development component requires one of the following:

- Windows NT Version 4.0 or later, running on an Intel® processor
- OS/2 Warp 4
- AIX 4.1.4
- Sun Solaris 5.5.1
- OS/390 1.3

The workstation on which the IMS Web Development component is installed must have:

- For NT, Microsoft Visual C++ 4.2 or later
- For AIX, IBM C Set++ for AIX
- For OS/2, IBM VisualAge C++ for OS/2
- For Sun, SPARCompiler C++
- For OS/390, IBM C/C++ Compiler for MVS/ESA

IMS Web Studio requires:

- The JDK or JRE 1.1.3 or later. To download the JDK or JRE, go to the following url: www.javasoft.com/products/

IMS Web Runtime component

The IMS Web Runtime component is installed on a Web server machine and requires:

- One of the following:
 - Windows NT Version 4.0 or later, running on an Intel processor
 - OS/2 Warp 4
 - AIX 4.1.4
 - Sun Solaris 5.5.1
 - OS/390 1.3

- A Web server product. IMS Web has been tested with the following Web server products:
 - IBM Internet Connection server on Windows NT, OS/2, AIX, Sun Solaris, and OS/390
 - Microsoft Internet Information server on Windows NT
 - Netscape FastTrack server on Windows NT

Net.Data

This section includes prerequisites for Net.Data and installation instructions.

Prerequisites

If you plan to use Net.Data as a front end for your IMS transactions, the following component is a prerequisite for Net.Data:

- IMS Web Runtime component installed on your Web server

Installation information for the IMS Web language environment

The first release of Net.Data (which included the IMS Web language environment) contained the IMS Web language environment at the IMS Web 1.2.0 level (DTWHWS.dll). This language environment only ran on a Windows NT Web server and supported nonconversational transactions. The IMS Web language environment at the IMS Web 2.1.0 level (DTWHWS2.dll) has enhancements that include support for conversational transactions.

Restrictions:

- DTWHWS2.dll requires the IMS Web Runtime Component at the 2.1.0 level.
- The IMS Web language environment of Net.Data is only supported when Net.Data runs as a CGI application. It is not supported with Net.Data as ICAP.

If you want to use the 2.1.0 level of the IMS Web language environment before the release of Net.Data that includes it, you must go to the IMS Web download page and proceed as follows:

1. Download the 2.1.0 IMS Web language environment (DTWHWS.dll) and place it in the same directory as the other Net.Data language environments. Typically, this directory is c:\winnt\system32.
2. Install the 2.1.0 IMS Web Runtime component on the Net.Data machine using the 2.1.0 IMS Web language environment. The IMS Web Runtime component is a prerequisite of the IMS Web language environment.

Chapter 3. IMS Web Development

This section describes the IMS Web Development process and includes the following topics:

- “How to install the IMS Web Development component”
- “Starting IMS Web Studio” on page 8
- “Creating a project in IMS Web Studio” on page 8
- “Associating MFS source with IMS Web Studio project” on page 9
- “Parsing files” on page 11
- “Defining transactions” on page 12
- “Defining classes” on page 13
- “Defining hosts” on page 15
- “Defining platforms” on page 16
- “Generating classes” on page 17
- “Directory structure and associated files” on page 17
- “Building an executable file” on page 19
- “IMS Web transactions” on page 20
- “Notes for version 1 users” on page 24
- “Java virtual machine considerations” on page 25

The IMS Web Development Component consists of the IMS Web Studio tool and a set of library and header files that are required to build the generated code into an executable that runs in the IMS Web Runtime environment. The IMS Web Development component lets you compile and link the generated C++ project files for the NT, OS/2, AIX, Sun, and OS/390 platforms.

How to install the IMS Web Development component

Requirement: JDK or JRE release 1.1.3 or later is a prerequisite. You can download the JDK or JRE by linking to the following url:

www.javasoft.com/products/

1. Select the Download link for the IMS Web Development. This connection downloads an executable file to your workstation.
2. Install JRE Version 1.1.4 (or higher) or JDK Version 1.1.3 (or higher). Ensure that you can run either **jre.exe** or **java.exe** from the command prompt. It is also helpful to add .; to your classpath to ensure that Java searches the current directory.
3. From the MS-DOS command line, change to the directory where the image (hwsjd21b.class) is located and enter:

```
jre -cp . HWSJD21B
```

Recommendation: If you are using the JDK, enter `java HWSJD21B`. However, we recommend that you use the JRE.

Starting IMS Web Studio

To start IMS Web Studio, select the Start button on your Windows 95 or Windows NT screen, and then do the following:

For Windows NT:

1. Select the **Programs** option from the popup menu.
2. Select the **IMS Web Studio V2** option.

The IMS Web Studio window opens, providing the options that you use to create a project.

You may navigate through the TaskGuide using Next/Back or with the buttons on the left of the TaskGuide. The tasks are checked off as you complete a task.

Note: For GA, we will add instructions for starting the tool on other platforms.

Creating a project in IMS Web Studio

A project is a container for organizing the files required to execute a single IMS transaction using IMS Web.

To create a new project, perform the following actions:

1. Select **Choose Project** from the IMS Web Studio tasks on the initial TaskGuide window or click the Next button. The Choose project TaskGuide window displays.

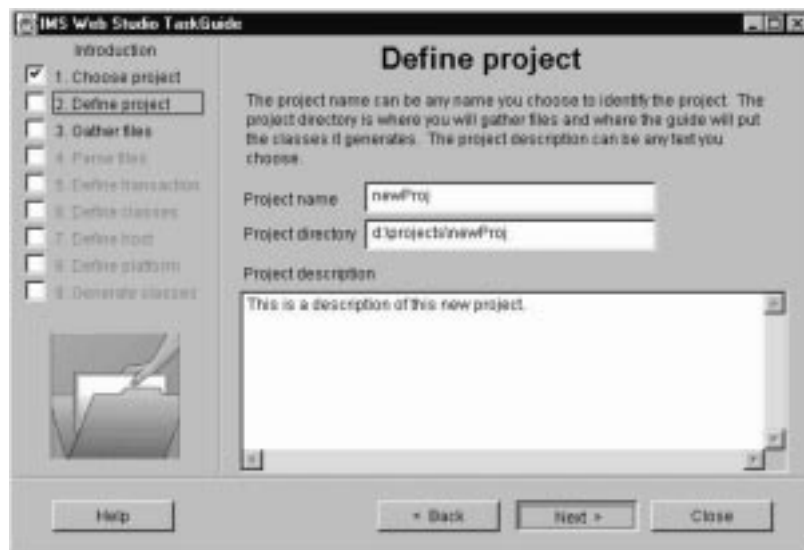


2. Click the Create new project button.



Then, click the Next button.

3. Specify project-related information in the Define project TaskGuide window, as in the following illustration.



Click the Next button. The Gather files TaskGuide window displays.

Associating MFS source with IMS Web Studio project

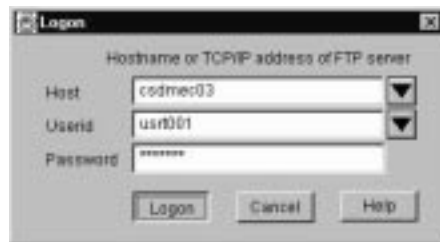
After you have created a project in IMS Web Studio, select and download the MFS source files that are associated with the transaction for which you want to generate code. To select and download these files, do the following:

1. To download files from the host, click the Download from host button.



A Logon dialog box displays.

2. Enter the host name of the machine where the MFS source library is located, enter your userid and password, and click the Logon button.



A download dialog box displays.

3. Enter a fully qualified data set name in the Fully qualified MFS dataset name field.
4. If you want to list all members beginning with cdc, enter a mask in the Dataset member mask field. (For example, enter "cdc*" to list all members beginning with cdc.)
5. Click Apply. Members of your data set display in the list box.
6. Select each member you want included and click the Download selected button. Then, click Close on the Download window, followed by clicking Next on the TaskGuide window. The Parse files TaskGuide window displays.

Notes:

- a. From the Gather files TaskGuide window, you may also use the Find MFS file on workstation button to move MFS files from another project directory on the workstation.
- b. From the Gather files TaskGuide window, you may also copy MFS files to your project directory (using whatever disk utility you normally use) and click the Refresh list button to reflect the current contents of the project directory.
- c. From the Gather files TaskGuide window, you may delete MFS files from the project directory using the Delete selected button.



Parsing files

Once the MFS source files are downloaded, they are parsed to identify the logical page (LPAGE) and message field (MFLD) entries that describe the input and output of your IMS transaction. This parsed information is used to generate HTML, C++ source code, and a Net.Data macro. This task is mandatory and must be completed each time you use the IMS Web Studio TaskGuide. After you complete this task, subsequent tasks are checked off if the TaskGuide has sufficient information to complete them without your interaction.



1. Make the following selections:

Project files

You can edit a project file (MFS file) by selecting it in the Project files list box and then by clicking the Edit selected button.

Edit selected

Edit the file you selected in the Project files list box. If you have not specified a default editor, an Open file dialog box displays so that you can find the executable file that starts your editor.

Specify editor

If you want to use a particular editor to edit the selected file, click the Specify editor button. An Open file dialog box displays. You can find the

executable file that starts your editor. This editor then becomes the default editor. You need to specify an editor for each installation of IMS Web.

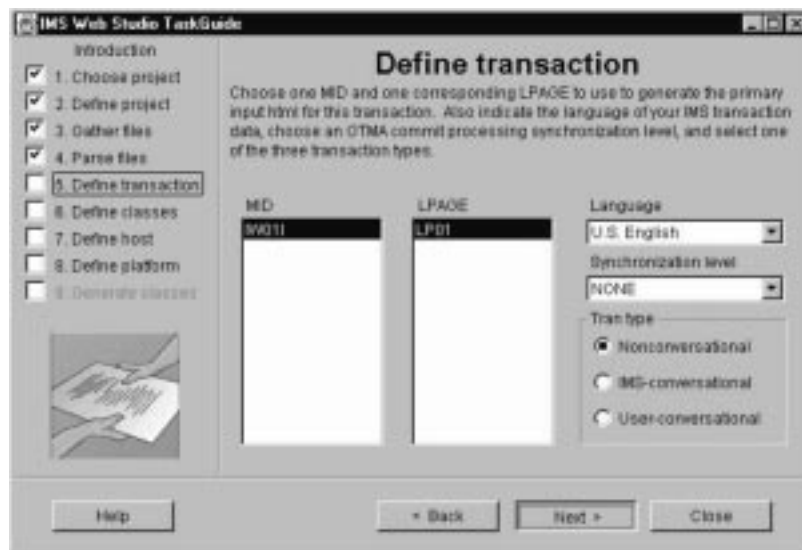
Parse all

Click the Parse all button when you are ready to have the project files parsed. Any syntax errors detected in the MFS source will be reported in an error dialog box. All MFS files in the project directory must be parsed successfully before you can proceed to the next tasks.

2. When you are finished, click Next.

The Define transaction TaskGuide window displays.

Defining transactions



1. Set up the transaction information needed to generate the primary input HTML.

MID The MIDs list box displays the names of all the message input descriptors in the MFS source associated with the project. If multiple MIDs are specified in the MFS source set, select a MID name from the list displayed.

LPAGE

The LPAGE list box displays the names of all the input logical page descriptors associated with the MID that is selected in the MIDs list box. If multiple LPAGES are specified in the selected MID, select an LPAGE name from the list displayed.

The MID and LPAGE are used as the primary input page for the transaction. For nonconversational transactions, this input page is the only input page. For conversational transactions, this input page is the first input page.

IMS Web transactions, whether conversational or nonconversational, are started from an HTML form that is generated from the single MID/LPAGE that is selected in the Define Transaction TaskGuide window. This MID/LPAGE typically contains an MFLD statement for the IMS transaction code.

For IMS conversational transactions, intermediate HTML forms are generated by combining the output fields of each MOD in the set of MFS source files parsed, and the input fields of the MID referenced by its NXT= keyword. In addition, IMS Web generates C++ classes for all the MIDs and MODs in the set of MFS source files being parsed.

Language

You are shown a list of the languages that IMS Web currently supports for IMS transaction data. For these languages, IMS Web converts the data between the host platform and server platform representations. Select the language of your IMS transaction data.

Synchronization level

Choose an OTMA commit processing synchronization level from the drop-down list. The default is NONE.

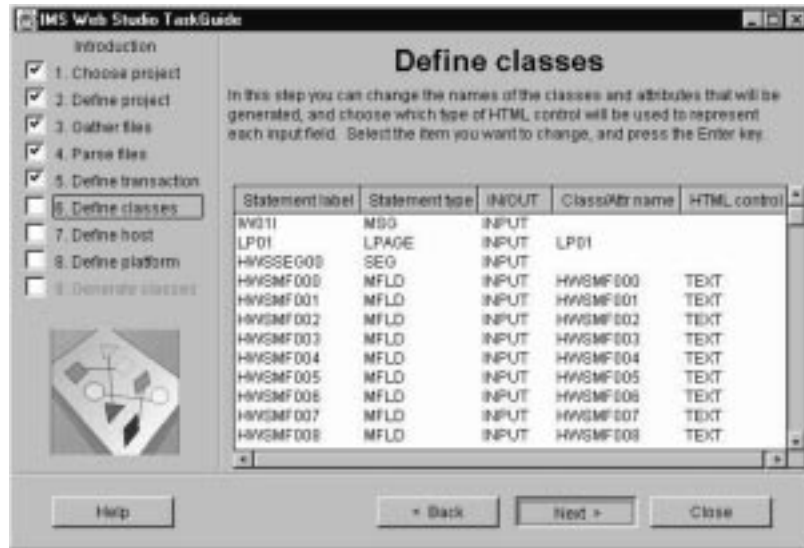
Tran type

Check the appropriate box depending on whether or not your transaction is nonconversational, IMS-conversational, or user-conversational.

Note: User-conversational transactions are a category of IMS transactions that have a continuous dialogue with a terminal without using IMS's SPA, so to IMS, they are not conversational. IMS Web Studio generates different code to support this type of transaction.

2. After making your selections, click the Next button. The Define classes TaskGuide window displays.

Defining classes



1. Define the class and attribute names to be generated.

Statement label

If the MFS statements have a label, they appear in this column. If they do not have a label, one is generated and appears in this column.

Statement type

The type (MSG, LPAGE, SEG, or MFLD) of statement appears in this column.

IN/OUT

Input represents an input message. Output represents an output message.

Class/Attr name

After the MFS source files are parsed, the LPAGE and MFLD names in the source, or those that IMS Web defaults, are loaded into the Class/Attr name list box on the Define classes page.

To edit a name, double-click the name in the Class/Attr Name list box. A dialog box displays so you can enter a new name.

Restriction: Don't use names beginning with HWS (HWS is reserved for IMS Web).

2. Click the Next button. IMS Web checks the names in the Class/Attr Name list box for duplication. Each LPAGE name and each MFLD name within each LPAGE name must be unique. If IMS Web finds a duplicate name, a message box displays, stating the nature of the problem. The second occurrence of the duplicated name is automatically selected for editing. Due to limitations of the list widget being used, the selected item cannot be automatically scrolled into view. Use the scroll bar to find the selected name.

Notes:

1. The names associated with LPAGE statements will become class names in the generated code, so be sure to use only valid C++ class names. Duplicate names will not be allowed, because it does not make sense to have non-unique class names in a single application.
2. The names associated with MFLD statements become member variable (attribute) names within the class named by the preceding LPAGE statement entry in the list. These names must be unique within each class or LPAGE, because it does not make sense to have two member variables with the same name in a single class.

HTML control

This represents the type of HTML control used for each input field. Currently, we only generate text fields.

The Define host TaskGuide window displays.

Defining hosts

The screenshot shows the 'Define host' dialog box in the IMS Web Studio TaskGuide. The window title is 'IMS Web Studio TaskGuide'. On the left, there is a 'Introduction' pane with a list of steps: 1. Choose project, 2. Define project, 3. Gather files, 4. Parse files, 5. Define transaction, 6. Define classes, 7. Define host (highlighted), 8. Define platform, and 9. Generate classes. Below the list is a small icon of a computer monitor. The main area is titled 'Define host' and contains the following text: 'In this step you can specify information about your transaction's interface to the host IMS. Values you specify will be used as defaults in your web application. You may also check whether or not you wish the input HTML to contain prompts for these fields.' There are two checkboxes: 'Prompt for host information' (checked) and 'Prompt for user information' (checked). Below these are four dropdown menus: 'Hostname of the machine running ITOC' (value: esdmac11), 'Port number' (value: 9999), 'Datastore ID/IMS XCF member name' (value: D9001), 'RACF group name' (value: HARRY), and 'RACF userid to pass to OTMA' (value: OOFISHIN). At the bottom, there are buttons for 'Help', '< Back', 'Next >', and 'Close'.

1. Enter the required information as follows:

Prompt for host information

If you check this box, the initial input HTML form will be generated with text entry fields (for Hostname, Port number, and Datastore ID/IMS XCF member name), and any values entered into the corresponding Define host TaskGuide window fields will be used as defaults. If you do not check this box, you must enter values into the Define host TaskGuide window fields and these values will be hard coded into the C++ source of the generated CGI-BIN program or transaction DLL.

Hostname of the machine running ITOC

The TCP/IP host name of the machine running the IMS TCP/IP OTMA Connection. Do not enter the TCP/IP address here, because the CGI will not work. The host name might or might not have to be fully qualified, depending on how TCP/IP is configured for your web server.

Port number

The port number (maximum of 4 characters) assigned to the IMS TCP/IP OTMA Connection on the machine running the IMS TCP/IP OTMA Connection

Datastore ID/IMS XCF member name

The extended coupling facility member name (maximum of 8 characters). This is the Datastore ID you specified when you installed IMS TCP/IP OTMA Connection.

Prompt for user information

If you check this box, the initial input HTML form will be generated with text entry fields (for RACF group name and RACF userid), and any values entered into the corresponding Define host TaskGuide window fields will be used as defaults. If you do not check this box, you must enter values into the Define host TaskGuide window fields and these values will be hard coded into the C++ source of the generated CGI-BIN program or transaction DLL.

RACF group name

The RACF group name (maximum of 8 characters) to be passed to IMS OTMA.

RACF userid to pass to OTMA

The RACF user ID (maximum of 8 characters) to be passed to IMS OTMA.

Prompt for password

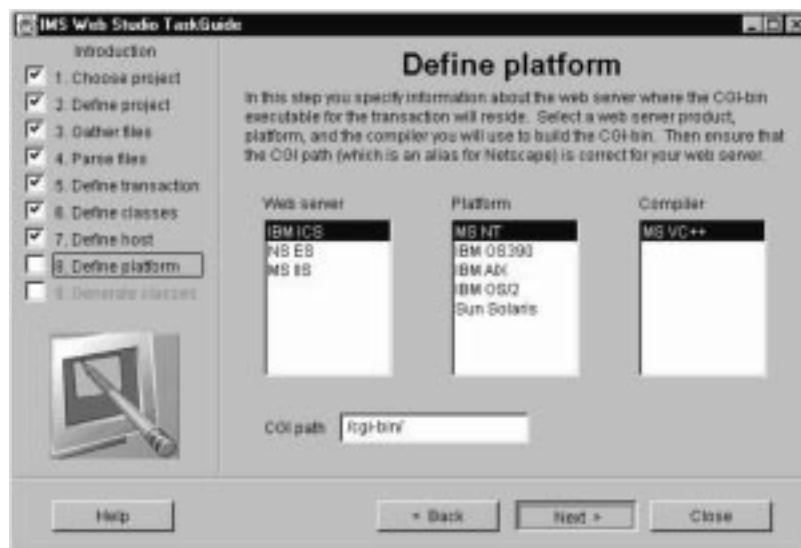
If you check the prompt for user information check box, the prompt for password check box is enabled. Otherwise, it is disabled.

If you check the prompt for password box, the initial input HTML form will be generated with a text entry field for password.

If the prompt for password check box is not checked, the password field is not generated.

2. Click the Next button. The Define platform TaskGuide window displays.

Defining platforms



1. Select and enter the required information as follows:

Web server

All supported servers are listed. Select a Web server product.

Platform

Select the target platform for your Net.Data or CGI application. All supported platforms for the selected server are displayed.

Compiler

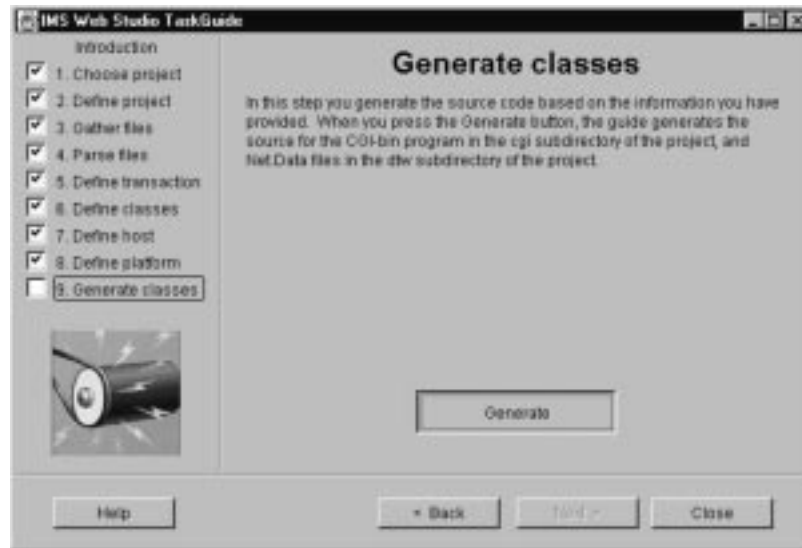
Select a compiler you will use to build the CGI-bin or Net.Data transaction DLL. This information is used to tailor the makefile for your compiler. If you use a different compiler, you might need to modify the generated makefile.

CGI path

The path where your Web server looks for CGI applications. The

supplied path name is the default path for the selected Web server. You can change this path to suit your installation.

2. When you are finished, click the Next button. The Generate classes TaskGuide window displays.



Generating classes

1. Click the Generate button to start the code generation.
2. After the classes have finished generating, click Close.

Directory structure and associated files

The following section outlines the directory structure and associated files for IMS Web. IMS Web Studio generates the CGI-BIN associated files and Net.Data files in two subdirectories of the IMS Web Studio project directory:

- cgi** For files associated with the submission of the IMS transaction using an IMS Web generated CGI-BIN program. The cgi subdirectory contains the .mak file for building the executable form of the CGI-BIN program.
- dtw** For files associated with the submission of the IMS transaction using Net.Data. The dtw subdirectory contains the .mak file for building the transaction DLL (the DLL or shared library that is dynamically loaded by Net.Data's IMS Web language environment).

The project directory contains the MFS source for the IMS transaction. The output from each .mak is placed in the corresponding subdirectory. The following are the files in each directory, where *proj* represents the first five characters of the IMS Web Studio project name.

- **proj**

- proj.hws**

- The IMS Web Studio project file.

proj.mfs

One or more MFS source files associated with the project. The name of the MFS source file *does not* have to be *proj*.

- **cgi**

cgiproj.cpp

The C++ source code for the CGI-BIN program that submits the IMS transaction. The CGI-BIN program uses additional classes generated in this subdirectory.

htmproj.htm

The input HTML form used to input data for the IMS transaction.

htmparse.cpp

The C++ source (implementation file) for the class that parses the input string from the generated input HTML form.

htmparse.hpp

The interface file for *htmproj.cpp*.

lpageIn.cpp

The C++ source (implementation file) for the class that represents the input logical page of the input message. This class includes a method that is used to submit the request (input message) to IMS and return one of the generated classes for the output logical pages.

lpageIn.hpp

The interface file for *lpageIn.cpp*.

lpageOut.cpp

The C++ source (implementation file) for an output logical page of an output message. A class is generated for each output logical page. This class includes a method that builds a buffer of HTML that is returned to the Web client.

lpageOut.hpp

The interface file for *lpageOut.cpp*.

etc. A nonconversational transaction can have cpp/hpp files for multiple output logical pages. A conversational transaction can have cpp/hpp files for multiple input and output logical pages.

cgiproj.mak

The makefile used to compile and link the source code, *cgiproj.cpp*, into an executable CGI-BIN program on the target platform.

- **dtw**

dtwproj.cpp

The C++ source code for the DLL or shared library that submits the IMS transaction. The transaction is invoked by Net.Data's IMS Web language environment.

dtwproj.def

The module definition file used in building the transaction DLL.

dtwproj.mac

The Net.Data Web macro that contains the input and output HTML for the IMS transaction.

dtwproj.htm

A sample link that invokes Net.Data as a CGI-BIN program and references the generated Web macro and input HTML form.

lpageIn.cpp

The C++ source (implementation file) for the class that represents the input logical page of the input message. This class includes a method that is used to submit the request (input message) to IMS and return one of the generated classes for the output logical pages.

lpageIn.hpp

The interface file for *lpageIn.cpp*.

lpageOut.cpp

The C++ source (implementation file) for an output logical page of an output message. A class is generated for each output logical page.

lpageOut.hpp

The interface file for *lpageOut.cpp*.

etc. A nonconversational transaction can have cpp/hpp files for multiple output logical pages. A conversational transaction can have cpp/hpp files for multiple input and output logical pages

dtwproj.mak

The makefile used to compile and link the source code, *dtwproj.cpp*, into a DLL or shared library on the target platform.

Building an executable file

IMS Web generates a .mak file that has the same name as the CGI-BIN program or Net.Data macro. For example, to build the CGI-BIN executable file *cgiproject_name.exe*, you run the generated .mak file *cgiproject_name.mak*. To build the transaction DLL *dtwproject_name.dll*, you run the generated .mak file *dtwproject_name.mak*.

In order to compile and link all the generated code for the project, you run the appropriate utility against the generated .mak file. For OS/2 and Windows NT, the utility is NMAKE; for AIX, Sun, and OS/390, it is MAKE. You run the utility on the platform for which you generated the source. You should use the Studio tool on your target platform, so you do not have to move the generated source before building the CGI executables.

The paths to the include and LIB files in the .mak file depend on where the IMS Web Development Component is installed when the code is generated. So it is best to use the Studio tool on the platform where you wish to build.

The generated .mak file depends on the macro CFG having been set. You can set CFG to either HWSDBG or HWSRe1, depending on whether or not you want your executable file built with debugging information.

Examples:

NMAKE /F *CGIproject_name.mak* CFG=HWSDBG

Compiles and links with debug information on OS/2 or Windows NT

MAKE -f *CGIproject_name.mak* CFG=HWSRe1

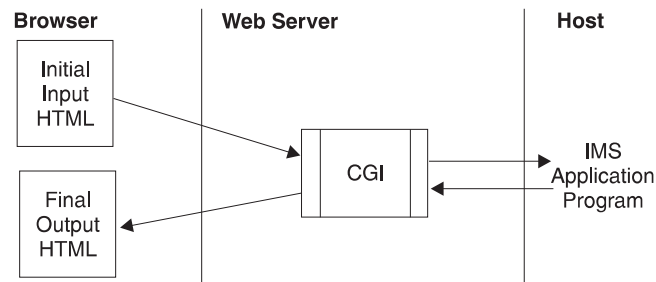
Compiles and links without debug information on AIX, Sun Solaris, and OS/390

IMS Web transactions

IMS Web supports three types of transactions: Nonconversational, IMS conversational, and user-conversational. This section describes all three types of transactions.

Nonconversational transactions

Nonconversational transactions are the simplest type of transaction supported by IMS Web. A single input to an IMS application program is followed by a single output. The following diagram illustrates how IMS Web supports this type of transaction.

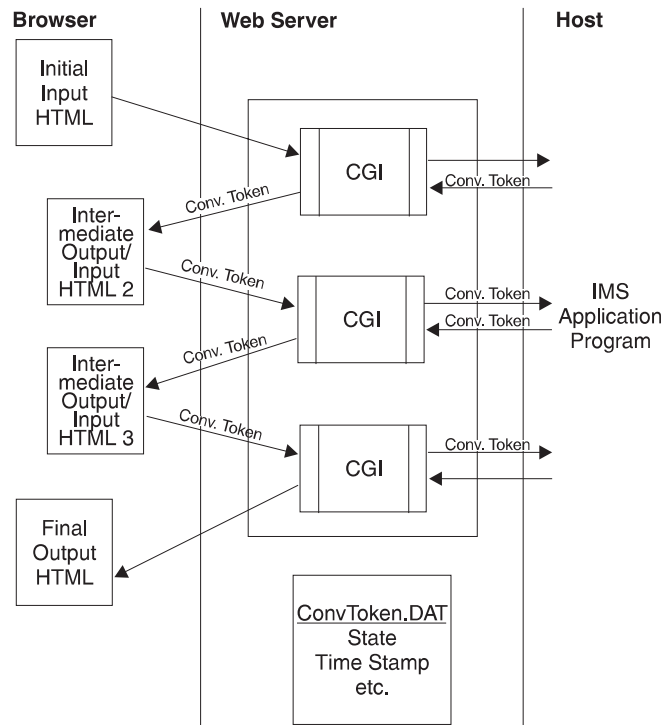


- The transaction is started by invoking the initial input HTML form. This form is generated based on the MFS MID chosen in the IMS Web Studio.
- When the CGI-BIN program associated with the transaction receives the output from the IMS application program, it returns the final output HTML form.

IMS conversational transactions

The IMS conversational transaction allows the end user to have a continuing dialogue with an IMS application program. The IMS application program uses the SPA to save state data for subsequent iterations of the conversation. The following

diagram illustrates how IMS Web supports this type of transaction.



- The conversation is started by invoking the initial input HTML form. This form is generated from an MFS MID chosen in the IMS Web Studio.
- When the CGI-BIN program associated with the transaction receives the output from the IMS application program, it saves state information about the conversation in a conversation state file, ConvTokn.DAT. In addition, it returns the Intermediate Output/Input HTML 2 form, along with conversation state data in hidden HTML fields of the form.

The Intermediate Output/Input HTML 2 form, and all intermediate HTML forms for a conversational transaction, are generated by combining the output and input fields of each MOD and the MID referenced by its NXT= keyword.

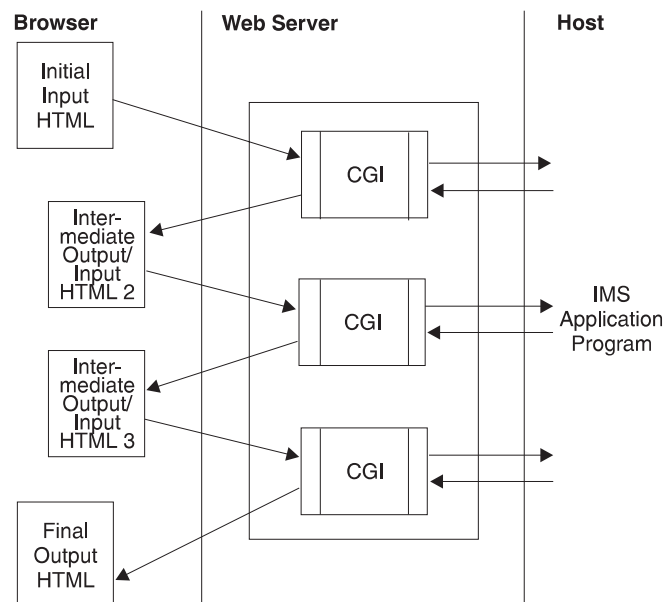
- When the Intermediate Output/Input HTML 2 form triggers the CGI-BIN program again, the CGI-BIN program gets ConvTokn from the hidden HTML fields and reads the ConvTokn.DAT conversation state file to validate the current state of the conversation.
- Iterations of the conversation continue. Each execution of the CGI-BIN program updates the conversation state file with new information about the state of the conversation.
- When the CGI-BIN program receives a message from the IMS application program indicating the end of the conversation, it deletes the ConvTokn.DAT file and returns the Final Output HTML form. The Final Output HTML form is an output only HTML form, containing the contents of the final message from the IMS application program.
- Intermediate HTML forms for a conversational transaction contain an additional button that can be used to end a conversation. Selecting this button causes the CGI-BIN program to erase the conversation state file, ConvTokn.DAT, and send a message to the host to end the conversation. An indication that a message was sent to the host is displayed at the browser.

User-conversational transactions

User-conversational transactions allow a customer to appear to have a continuing dialogue with one or more IMS application programs. The IMS application program(s) *do not* use the SPA to save state data for subsequent iterations of the conversation. It is up to the user to save state data between iterations, and IMS does not consider the transaction to be conversational.

The following diagram illustrates how IMS Web supports this type of transaction.

Exception: This type of conversation is not supported by the Beta level of IMS Web 2.1.0.



- The conversation is started by invoking the Initial Input HTML form. This form is generated from an MFS MID chosen in the IMS Web Studio.
- When the CGI-BIN program associated with the transaction receives the output from the IMS application program, it returns the Intermediate Output/Input HTML 2 form. The Intermediate Output/Input HTML 2 form, and all intermediate HTML forms for the user-conversational transaction, are generated by combining the output and input fields of each MOD and the MID referenced by its NXT= keyword.
- The saving of conversation state information at the host is under total control of the IMS application program.
- When the CGI-BIN program receives a message from the IMS application program indicating that a MOD with no NXT= keyword be used for formatting, the CGI-BIN program generates an output only HTML form, Final Output HTML, rather than a form, to terminate the conversation.

IMS Web protocol for conversations

As mentioned previously in this section, state information about a conversation is saved in hidden fields of intermediate HTML forms of the conversation. For this reason, the input data for successive iterations of a conversation must be provided in the appropriate intermediate HTML forms. You **may not** use the BACK function to

select an HTML form for use as input to a conversation in progress. If you do, a protocol error message will be presented at the browser and the conversation will be terminated.

There are a number of situations where an active IMS Web conversation can be *orphaned*. An orphaned conversation is still active from the point of view of IMS Web and IMS, but it can no longer be continued or terminated using the browser that initiated it. An orphaned conversation can be terminated using the IMS Web command, HWSCMD EXIT.

Recommendation: Orphaned conversations can occur when a browser session is closed while an IMS Web conversation is in progress in the browser session. In addition, we **strongly** recommend that a single IMS Web-initiated conversation be conducted in a single browser session. Multiple IMS Web conversations in a single browser session involve the BACK function, and can very easily result in orphaned conversations or protocol errors.

IMS conversational transactions

This section describes how to display and terminate conversational transactions.

Displaying conversational transactions: The IMS command, /DISPLAY CONVERSATION BUSY is typically used to display currently active conversations. Using this command will display both conversations initiated by IMS Web and those initiated from an IMS terminal. For conversations initiated by IMS Web, /DISPLAY CONVERSATION BUSY output is:

TERMINAL

The OTMA TPIPE name, which is specified to the IMS Web Studio tool as *Datastore ID/IMS XCF member name*.

USER The userid associated with the IMS Web-initiated conversational transaction and specified to the IMS Web Studio tool as *RACF userid to pass to OTMA*.

nnnn An identifier of the conversation as known by IMS. This identifier represents the *host end* of the IMS Web-initiated conversation.

IMS Web-initiated conversations can be displayed using the IMS Web Runtime command, HWSCMD DISPLAY CONV. Using this command displays only IMS Web-initiated conversations; using this command does not display conversations initiated from an IMS terminal. The output for this command is:

Conversation token (Conv.Token)

The IMS Web identifier for the conversation. This identifier is formed by converting the second half of the ServerToken field in the OTMA message into an 8-character identifier. ServerToken is OTMA's identification of the conversation. The first half of ServerToken is a STCK value; the second half is the STCK value converted into a readable date/time value. The conversation token is used as the filename of the conversation state file, ConvTokn.DAT.

RUName

Request Unit Name. RUName is a unique name that identifies the request the IMS Web program makes to execute the IMS transaction. IMS uses this name as the name of the LTERM that is associated with the transaction.

Next Input Lpage

This is the value of the NXT= keyword of the message output descriptor (MOD) that was used to format the output of the current iteration of the conversation.

Date/Time Stamp

This is the date and time the current iteration of the conversation was presented to the customer. The date and time is updated by IMS Web Runtime, for each iteration of a conversation, when an output message is received from the IMS application program. The output message is then presented to the customer in an HTML form.

Age (min.)

The length of time since the latest iteration of the conversation was presented to the customer. It is the difference between the current date and time and the date/time stamp, in minutes.

IMS Name

The XCF member name for IMS OTMA. The output of the IMS command, /DISPLAY CONVERSATION BUSY displays this name in the TERMINAL field for IMS Web-initiated conversations.

Host Name

The TCP/IP host name of the machine on which the IMS TCP/IP OTMA Connection is installed.

Terminating conversations: You cannot use the IMS command, /EXIT to terminate conversations initiated by IMS Web. IMS Web-initiated conversations are terminated either by the IMS application program, or by selecting the TERMINATE button on the IMS Web HTML form. You can also terminate IMS Web conversations by using the IMS Web Runtime command, HWSCMD EXIT CONV; however, reserve this method for unusual situations.

User-conversational transactions

Restriction: This type of conversation is not supported by the Beta level of IMS Web 2.1.0.

Notes for version 1 users

- Using existing projects:
 - You may create a new project using the version 2 Studio tool in an existing version 1 project directory. The MFS source in the existing project will be recognized by the version 2 TaskGuide.
 - You may create a new project in a separate directory and copy the MFS files from the existing project into the new project directory. Use the Refresh list button on the Gather files TaskGuide window to reflect the current contents of the project directory.
 - The version 2 parser is more strict regarding MFS syntax. So, it is possible that MFS files that were parsed successfully by the version 1 tool will fail with the version 2 syntax checker.
- Generating on your target platform
 - In version 1, the Studio tool was only available on Windows NT. As a result, CGI and Net.Data source generated for other platforms had to be moved from NT to the target platform before it could be built. The version 2 Studio tool will become available on all supported target platforms as the version 1.1.3 Java virtual machines become available on these platforms. This will allow you to

generate the source on the same machine where it will be built, simplifying the overall process. Currently, the .mak files generated by the Studio tool are platform specific in how they locate the include and LIB files. So, it is recommended that you generate on the target platform in order to reduce the need to modify the generated .mak files.

- Redesigned Studio tool
 - The addition of conversational transaction support, and the desire to provide the Studio tool on multiple platforms, necessitated a complete overhaul of the Studio tool. While most of the existing concepts remain the same, new features have been added to the tool. In addition, some fields have been renamed as a result of customer feedback. Please thoroughly read this user guide to understand how to use version 2.

Java virtual machine considerations

Attention: Be sure to read this section before you change your Java virtual machine (JVM).

When you install the IMS Web Studio version 2, a launch script is generated that you use to start the Studio tool, and an uninstall script is generated that can be used to remove the Studio tool from your workstation. These scripts contain calls that are dependent on the location of the JVM used to run the installation. If you remove or replace the JVM on your workstation, these scripts will probably no longer function. To avoid this situation, uninstall the Studio tool before changing your JVM.

If you have already changed the JVM, and are reading this because the launch and uninstall scripts no longer work, reinstall the Studio tool using the new JVM to update the launch and uninstall scripts.

Chapter 4. IMS Web Runtime component

This section describes the IMS Web Runtime component and the following topics:

- “How to install the IMS Web Runtime component”
- “Incorporating project executable files into your Web server environment” on page 29
- “IMS Web Runtime commands” on page 30
- “IMS Web Runtime data conversion” on page 34

The IMS Web Runtime component enables you to run the project executable files on the Windows NT, OS/2, AIX, Sun, or OS/390 platforms. The Runtime component consists of:

- The IMS Runtime DLLs (or shared files for AIX and Sun) required to run an IMS Web CGI-BIN or Net.Data transaction DLL
- The conversion tables for IMS Web supported languages
- An executable file (hwscmd.exe) that is used to execute IMS Web commands on the Web server machine.

Attention: It is possible to install the runtime and development components on different machines provided that both machines run the same operating system.

How to install the IMS Web Runtime component

Go to the IMS Web page download and select the GA versions. Enter your password and click on the submit button or fill out the survey and click the send survey button. Then, proceed as follows.

For a Windows NT Web server platform:

1. Select the Download link for the IMS Web Runtime component for Windows NT Web servers. This connection downloads an executable file to your workstation.
2. Using the Save dialog box, save the downloaded file to a temporary directory.
3. Run the downloaded file (which invokes Installshield to install the IMS Web Runtime component) on the Windows NT Web server platform. Follow the installation script that is provided.

For an OS/2 Web server platform:

1. Select the Download link for the IMS Web Runtime component for OS/2 Web servers. This connection downloads an executable file to your workstation.
2. Using the Save dialog box, save the downloaded file to a temporary directory.
3. Run the downloaded file on the OS/2 Web server platform in the temporary directory. The executable file expands into multiple files in the temporary directory.
4. Enter INSTALL from the command line of the temporary directory to invoke Software Installer to install the IMS Web Runtime component. Follow the installation script that is provided.

For an AIX Web server platform:

1. Select the Download link for the IMS Web Runtime component for AIX Web servers. This connection downloads an image file (HWSXBvrm.img) to your workstation. The "vrm" in the file name refers to the version, release, and modification of the file.
2. Ensure that you are the root user, and then perform the following actions:
 - a. Type `smi t` to bring up the System Management Interface Tool.
 - b. Select the **Software Installation and Maintenance** option.
 - c. Select the **Install and Update Software** option.
 - d. Select the **Install/Update Selectable Software** option.
 - e. Select the **Install Software Products at Latest Level** option.
 - f. Select the **Install New Software Products at Latest Level** option.
 - g. In the dialog box that displays, specify the directory where you downloaded the image file. Select the OK push button.
 - h. When the multi-select list displays, select the OK push button.
 - i. When the "Are you sure?" dialog box displays, select the OK push button to start the installation.

For a Sun Web server platform:

1. Select the Download link for the IMS Web Runtime component for Sun Web servers. This connection downloads a package file (HWSSBvrm.pkg) to your workstation. The "vrm" in the file name refers to the version, release, and modification of the file.
2. Using the Save dialog box, save the downloaded file to a temporary directory called `/tmp`.
3. Log in as root.
4. Enter `pkgadd -d /tmp/HWSSBvrm.pkg IMSWeb`

For an OS/390 Web server platform:

See "Considerations when using ICS for OS/390" on page 34.

1. Before beginning this procedure, you should work with your OS/390 OE system administrator to ensure that your userid has the appropriate authorization to install the IMS Web Runtime DLLs and the HWSCMD utility on your OS/390 system.
2. Select the Download link for the IMS Web Runtime component for OS/390 Web servers. Selecting this link downloads the file, HWS3Bvrm.exe, to your workstation (where the "vrm" in the file name refers to the version, release, and modification of this file).
3. Run the downloaded .exe file, which is a self-extracting, compressed file, by entering `HWS3Bvrm install_directory` at a DOS command prompt (where again vrm represents the version, release, and modification of IMS Web and install_directory represents a valid DOS path that specifies the directory into which the uncompressed files will be extracted; use '.' (a single period) to represent the current directory). Execution of this file generates a tar file, HWS3Rvrm.tar, which contains the IMS Web Runtime component, along with HTML and text versions of both these installation instructions and a list of changes to the contents the tar file.
4. Upload the HWS3Rvrm.tar file to the `/usr/lpp/imsweb` directory in the HFS on the appropriate OS/390 OE runtime machine(s). Be sure that you set the transfer mode to binary and do not use any character conversion since the tar file has been created on an OS/390 machine running OE.

5. Unpack HWS3Rvm.tar. This unpacks the IMS Web runtime DLLs (used when your project CGI-BIN and/or Net.Data DLL files are executed) into the /usr/lpp/imsweb/lib directory, and unpacks the hwscommand executable file (used to start and stop the IMS Web Runtime tracing) into the /usr/lpp/imsweb/bin directory. In addition, the conversion tables for the IMS Web supported languages are unpacked into the /usr/lpp/imsweb directory. Again, no character conversion should be used when you unpack HWS3Rvm.tar because the characters are already in EBCDIC format.

Incorporating project executable files into your Web server environment

To incorporate the executable files that you have created in your IMS Web projects into your Web server environment, perform the following steps:

For IMS Web only

1. Copy the CGI-BIN executable and the generated HTML file to your Web server.
 - Place the CGI-BIN executable file in the Web server's CGI-BIN directory (Web Server's CGI Path). This directory is the one in which your Web server looks for CGI-BIN scripts and might be a relative directory name or an alias.
 - Place the HTML file in your Web server's HTML directory.
2. Optionally, you may add a link to the input HTML form in the appropriate HTML file(s) in your Web server HTML file tree. You can then use the link to display the input HTML form for the transaction on a Web browser.
 - Use this link or enter the URL directly in your browser to display the HTML form on your Web server.
 - Fill in the transaction input and select the SUBMIT push button on the form to submit the transaction to IMS and receive its output at the Web browser.

For Net.Data

1. Copy the transaction DLL (DTWproj.dll) and Net.Data Web macro (DTWproj.mac) to your Web server.
 - Place the macro in a directory from which Net.Data retrieves Web macros. (See the MACRO_PATH statement in Configuring Net.Data in the *Net.Data Administration and Programming Guide*.)
 - Place the transaction DLL in a directory from which the operating system retrieves DLLs or shared libraries.
2. Optionally, you may use the link in the sample file, DTWproj.htm, to modify an HTML file in your Web server's HTML tree. You can then use the link to invoke Net.Data and display the input HTML form for the transaction on a Web browser.
 - Use this link or enter the URL directly in your browser to display the HTML form on your workstation.
 - Fill in the transaction input and select the SUBMIT push button on the form in order to submit the transaction to IMS and receive its output at the Web browser.

IMS Web Runtime commands

IMS Web provides a number of commands to help you control IMS Web conversations as well as diagnose problems in the Runtime component. The command processor executable, `HWSCMD`, is invoked with appropriate parameters, from any command prompt on the Web server. The parameters you specify to the command processor allow you to:

- Display currently active IMS Web conversations
- Terminate IMS Web conversations
- Trace the execution flow of the Runtime component

Note: The syntax notation used for IMS commands is also used for IMS Web Runtime commands.

HWSCMD TRACE command

This command is used for tracing the execution of a CGI-BIN program or transaction DLL through the IMS Web Runtime component. This trace is primarily for diagnostic use by IBM service personnel.

A temporary trace file, *program_name.tmp*, records sequential executions of a CGI-BIN program or transaction DLL. *Program_name* is the name of the CGI-BIN program or Net.Data transaction DLL during tracing. When the trace is turned off, a final entry is made in the .tmp file and the file is renamed to *program_name.log*. These files can be found in the trace subdirectory of the IMS Web Runtime installation directory for Windows NT and OS/2, in `/usr/lpp/imsweb/trace/` for AIX and OS/390, and in `/opt/imsweb/trace/` for Sun.

The IMS Web Runtime component trace is intended primarily for diagnosis. For this reason, only one instance of a particular CGI-BIN program should be traced at a time. Trace data consists of the following:

- An entry that indicates the start of tracing for the CGI-BIN program or transaction DLL
- Date/time-stamped entries that represent the flow through the IMS Web Runtime component for each sequential execution of the CGI-BIN program or transaction DLL
- An entry that indicates the end of tracing for the CGI-BIN program or transaction DLL

Trace data is appended to an existing *program_name.tmp* file. If no *program_name.tmp* file exists, a new file is created containing the trace data.

Command syntax

```
▶▶ HWSCMD TRACE [ON|OFF] prog_name ▶▶
```

HWSCMD Is the name of the command processor executable.

Note: For AIX, Sun Solaris, and OS/390 OpenEdition Web server platforms, the command processor executable (`hwscmd`) must be entered in lower case.

Parameters

TRACE Specifies the trace function of hws cmd.

ON Starts the IMS Web Runtime component trace for the specified CGI-BIN program or Net.Data transaction DLL. The trace remains on for all future executions of the CGI-BIN program or Net.Data transaction DLL, until it is turned off.

OFF Ends the IMS Web Runtime component trace for the specified CGI-BIN program or Net.Data transaction DLL. Ending the trace makes the trace log file available.

program_name

Specifies the name of the CGI-BIN program or Net.Data transaction DLL being traced. *program_name* is generated by IMS Web Studio.

Attention: Before you can run hws cmd trace on, you must have obtained full access permissions for the directory where the temporary trace file and the trace log file are stored. The CGI-BIN program or Net.Data transaction DLL must also possess write permission for the directory.

HWSCMD DISPLAY CONV command

This command displays the information in one or more conversation state files on the Web server. See “IMS Web transactions” on page 20 for a description of IMS Web’s support of conversational processing and a description of conversation state files. These files represent conversations initiated by IMS Web from the Web server, and are located:

- In the temp subdirectory of the IMS Web Runtime installation directory for Windows NT and OS/2
- In the /usr/lpp/imsweb/temp subdirectory for AIX and OS/390
- In the /opt/imsweb/temp subdirectory for Sun Solaris

HWSCMD DISPLAY CONV can be used to display:

- All active IMS Web conversations, or
- A specific IMS Web conversation, or
- All active IMS Web conversations whose last iteration occurred more than *mmm* minutes ago.

Command syntax



HWSCMD Is the name of the command processor executable.

Note: For AIX, Sun Solaris, and OS/390 OpenEdition Web server platforms, the command processor executable (hws cmd) must be entered in lower case.

Parameters

DISPLAY

Specifies the display function of hws cmd.

CONV Is used to display IMS conversations. CONV is required.

ALL Is used to display all active IMS Web conversations.

convToken

Is used to display a specific IMS Web conversation.

GTmmm

Is used to display IMS Web conversations whose last iteration occurred more than *mmm* minutes ago, where *mmm* must be in the range of 1 to 999.

HWSCMD DISPLAY command sample output

The following example shows output from the HWSCMD DISPLAY CONV ALL command entered on the Web server when two IMS Web conversations are in progress on the browser.

Conv.Token	RUName	Next Input LPage	Date/Time Stamp	Age (minute)	IMS Name	Host Name
1I09276J	HWSGJAER	HWSLPG01	02/05/98 09:10:09	8	SOCKEYE	CSDMEC13
1I09184Z	HWSGJ8F	HWSLPG01	02/05/98 09:08:24	10	SOCKEYE	CSDMEC13

Processing completed for HWSCMD DISPLAY CONV ALL 02/05/98 09:18:04

where:

- ConvToken is the IMS Web identifier for the conversation. This identifier is formed by converting the second half of the ServerToken field in the OTMA message into an 8-character identifier. ServerToken is OTMA's identification of the conversation. The first half of ServerToken is a STCK value; the second half of ServerToken is the STCK value converted into a readable date and time value. A conversation token is used as the filename of the conversation state file (ConvToken.DAT).
- RUName is Request Unit name. RUName is a unique name that identifies the request the IMS Web application makes to execute the IMS transaction. IMS uses this name as the name of the LTERM that is associated with the transaction.
- Next Input LPage is the value of the NXT= keyword of the message output descriptor (MOD) that was used to format the output of the current iteration of the conversation.
- Date/Time Stamp is the date and time the current iteration of the conversation was presented to the end user. Date/time is updated by IMS Web Runtime, for each iteration of a conversation, when an output message is received from the IMS application program. The output message is then presented to the end user in an HTML page.
- Age (minute) is the length of time since the latest iteration of the conversation was presented to the end user. It is the difference between the current date and time and the date and time stamp, in minutes.
- IMS Name is the XCF member name for IMS OTMA. The output of the IMS command /DISPLAY CONVERSATION BUSY displays this name in the TERMINAL field for IMS Web initiated conversations.
- Host Name is the TCP/IP host name of the machine on which the IMS TCP/IP OTMA Connection is installed.

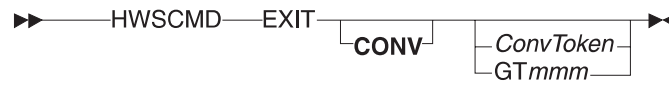
The following example shows output from the IMS /DIS CONV BUSY command when the same two IMS Web conversations are in progress. An additional conversation, initiated from an IMS terminal, is also in progress. This conversation is known to IMS, but not IMS Web.

```
00- 09.12.43 JOB00123 DFS000I  TERMINAL      USER      ID  STATUS - SYS3
- 09.12.43 JOB00123 DFS000I  SOCKEYE     GOFISHIN  0001 ACTIVE - SYS3
- 09.12.43 JOB00123 DFS000I  SOCKEYE     GOFISHIN  0002 ACTIVE - SYS3
- 09.12.43 JOB00123 DFS000I  T1303      USRT002   0003 ACTIVE - SYS3
- 09.12.43 JOB00123 DFS000I  *98036/091243*  SYS3
```

HWSCMD EXIT command

The HWSCMD EXIT command is used to terminate:

- A specific IMS Web conversation, or
- All conversations whose last iteration occurred more than *mmm* minutes ago



Command syntax

HWSCMD Is the name of the command processor executable.

Note: For AIX, Sun Solaris, and OS/390 OpenEdition Web server platforms, the command processor executable (*hwscmd*) must be entered in lower case.

Parameters

EXIT Specifies the exit function of *hwscmd*.

CONV Is used to terminate IMS Web conversations. *CONV* is optional.

convToken

Is used to terminate a specific IMS Web conversation.

GTmmm

Is used to terminate IMS Web conversations whose last iteration occurred more than *mmm* minutes ago, where *mmm* must be in the range of 1 to 999.

Normally, an IMS Web conversation is terminated by either the IMS application program or when the end user selects the TERMINATE button on an intermediate HTML form. However, if a browser session is closed while an IMS Web conversation is in progress, the conversation is still active and must be terminated using the HWSCMD EXIT CONV command. The HWSCMD EXIT CONV command may also be entered from the Web server while an IMS Web conversation is in progress on the browser. In this case, a protocol violation message is issued when the end user next selects the SUBMIT button.

The HWSCMD EXIT CONV command uses information in the conversation state file, *convToken.DAT*, to build a message and send it to IMS to end the conversation. HWSCMD EXIT CONV then deletes the conversation state file.

IMS Web Runtime data conversion

When data moves between products on different platforms, it might be converted from one representation to another. The representation of data on a particular platform is associated with a code page that reflects both the language of the data (e.g., U.S. English, Latin1, etc.) as well as the platform where the code page is used.

The data that you enter on a browser in the input HTML form generated by IMS Web Studio is converted by your Web server from the code page that is used by the browser to the code page that is used by the Web server. It is then converted by the IMS Web Runtime component from the Web server code page (see Note 1 on page 35) to the host code page used by IMS for data. After the transaction runs on IMS and a response is sent back, the code page conversion process is followed in reverse so that data in the response is eventually converted to the correct code page for the originating browser. This enables the response to be displayed correctly in the output HTML.

The code page conversion tables used by the IMS Web Runtime component are packaged with the component. The following tables list the code pages used by the IMS Web Runtime component, based on Web server platform and language.

Web Server Platform NT, OS/2, AIX, or Solaris

Language	Web Server Code Page	Host Code Page
U.S.English	IBM-1252	IBM-37
Latin1	IBM-1252	IBM-500
Latin2	IBM-1250	IBM-870
Hebrew	IBM-1255	IBM-424
Turkish	IBM-1254	IBM-1026

Web Server Platform OS/390

Language	Web Server Code Page	Host Code Page
U.S.English	IBM-1047	IBM-37
Latin1	IBM-1047	IBM-500
Latin2	IBM-870	IBM-870
Hebrew	IBM-424	IBM-424
Turkish	IBM-1026	IBM-1026

Considerations when using ICS for OS/390

When using the IMS Web Runtime component with ICS for OS/390, you must set the FsCp parameter in the ICS configuration parameters to the value for the Web server code page used by IMS Web in order to guarantee that all data will be converted correctly. See Note 2 on page 35. For example:

Language	FsCp	NetCp
U.S.English	IBM-1047	IBM-819

Language	FsCp	NetCp
Latin1	IBM-1047	IBM-819
Latin2	IBM-870	IBM-912
Hebrew	IBM-424	IBM-916
Turkish	IBM-1026	IBM-920

Notes:

1. This assumes that you have followed the recommendations in “Considerations when using ICS for OS/390” on page 34.
2. Some or all of your data may be converted correctly even if the FsCp setting in ICS for OS/390 does not match the Web server code page used by IMS Web (see “Web Server Platform OS/390” on page 34) since some of the conversion tables are identical except for a small number of characters.

Recommendation: However, it is **strongly** recommended that you set the FsCp in your Web server’s configuration parameters to match IMS Web’s Web server code page in order to guarantee that all data will be converted correctly.