CICS for MVS/ESA

CICS Web Interface Guide

Version 4 Release 1

SC33-1892-01

CICS for MVS/ESA

CICS Web Interface Guide

Version 4 Release 1

SC33-1892-01

Note!
Before using this information and the product it supports, be sure to read the general information under "Notices" on page v.

First Edition (November 1996)

This edition applies to the IBM licensed program CICS for MVS/ESA Version 4 Release 1, program number 5655-018, and to all subsequent versions, releases, and modifications until otherwise indicated in new editions. Consult the latest edition of the applicable IBM system bibliography for current information on this product.

Order publications through your IBM representative or IBM branch office serving your locality. Publications are not stocked at the address given below.

At the back of the publication is a page entitled "Sending your comments to IBM". If you want to make comments, but the methods described are not available to you, please address them to:

IBM United Kingdom Laboratories,
Information Development,
Mail Point 095,
Hursley Park,
Winchester,
Hampshire,
England,
SO21 2JN.

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Notices v
Programming interface information v
Trademarks and Service Marks vi

Summary of Changes vii
Changes to the first edition vii

Preface ix
What this book is about ix
Who this book is for ix
What you need to know to understand this book . . x
Notes on terminology x
Determining if a publication is current. x

Bibliography xi
CICS books for CICS for MVS/ESA xi
Obtaining the books xi
Other CICS books xi
HTTP and HTML information xi
HTTP xi
HTML xii
TCP/IP for MVS xii
Programming xii
C/370 xii
COBOL xii
PL/I xii
Assembler. xii
Language Environment/370. xii
Miscellaneous xii

Chapter 1. Introduction 1
CICS and the Internet 1
HTTP requests and responses. 3
CICS Web Interface transactions 4
Connection manager (CWBC). 4
Server controller (CWBM) 4
Alias (CWBA) 4
The CICS Web Interface data set. 5
User-replaceable programs. 5
The analyzer 5
The converter 5
Application programming with the CICS Web Interface 6
Control flow in an HTTP request 6
Data flow in an HTTP request 8
Dealing with non-HTTP requests 9
Implementing the CICS Web Interface 9

Chapter 2. Setting up the CICS Web Interface 11
Prerequisites for using the CICS Web Interface. . . 11
MVS/ESA 11
CICS for MVS/ESA. 11
TCP/IP for MVS 12
Installation process 12
Installing the CICS Web Interface with SMP/E . 12
Other installation tasks 12
Defining resources to CICS 13
Transaction definitions for extra alias transactions 13
Program definitions for user-replaceable programs 13
Transient data definitions. 14
Completing the CICS Web Interface data set file definition 14
Setting up a PDS for the template manager. . . 14
Defining a conversion table 15
Running the sample application 16
Changes to TCP/IP for MVS 16

Chapter 3. Writing user-replaceable programs for the CICS Web Interface . 19
Writing the user-replaceable programs 19
Writing an analyzer. 19
Writing a converter—general 23
Writing a converter—Decode 23
Writing a converter—Encode 25
Reference information for the analyzer 26
Analyzer 27
DFHWBADX responses and reason codes . . . 32
Reference information for the converter 32
Decode 34
Encode 38

Chapter 4. Writing CICS applications for the CICS Web Interface 41
Writing CICS programs to process HTTP requests 41
A sample application program 42
Reference information for the HTML template manager 42
Parameters in the communication area 43
Responses and reason codes. 46
Symbols, symbol table, and symbol lists. . . . 47
Reference information for the environment variables program 49
Reference information for the state management sample programs 51
Reference information for the parser program . . . 53
Unescaping considerations 54

Chapter 5. Configuration with the connection manager 55
What the connection manager is for 55
Enabling the interface 55
Disabling the interface. 55
Starting the connection manager 56
When the interface is disabled 56
When the interface is enabled 57
Using the connection manager BMS panels . . . 57

Starting the connection manager when the CICS Web Interface is disabled 58
Starting the connection manager when the CICS Web Interface is enabled 58
Enabling the CICS Web Interface 59
Setting and modifying options 59
Validating, saving, and activating options . . . 61
Disabling the CICS Web Interface 61
On CICS for MVS/ESA normal shutdown . . . 62
On CICS for MVS/ESA immediate shutdown . . 63
Updating the CICS Web Interface data set 63
Updating the CICS Web Interface status 64

Chapter 6. Security 65
Security for the CICS Web Interface 65
Security for the HTML template manager PDS . 65
Security for the CICS Web Interface transactions 65
The security sample programs 66

Chapter 7. Problem determination . . . 69
Recovery procedures 70
Product design considerations 70
Troubleshooting 70
Defining the problem 70
Documentation about the problem. 71
Using messages and codes 71
CICS Web Interface trace information. 72
CICS Web Interface trace points 72
Dump and trace formatting 82
Debugging the user-replaceable programs 82
Using EDF. 82
Using trace entries 82
Writing messages 83
Abends. 83

Appendix A. The business logic interface 85
A two-tier programming model. 85
What is the business logic interface? 86
Reference information 87
Business logic interface 88

Appendix B. Messages and Codes . . . 93

Index 117

Notices

The following paragraph does not apply to any country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact Laboratory Counsel, MP 151, IBM United Kingdom Laboratories, Hursley Park, Winchester, Hampshire, England SO21 2JN. Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577, U.S.A.

Programming interface information

This book is intended to help you use the CICS Web Interface support in CICS for MVS/ESA.

This book also documents Product-sensitive Programming Interface and Associated Guidance Information and Diagnosis, Modification or Tuning Information provided by CICS.

Product-sensitive programming interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning of CICS. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-sensitive programming interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Product-sensitive Programming Interface and Associated Guidance Information is identified where it occurs, by an introductory statement to a chapter or section.

Diagnosis, Modification or Tuning Information is provided to help you diagnose problems in your CICS system.

Note: Do not use this Diagnosis, Modification or Tuning Information as a programming interface.

Diagnosis, Modification or Tuning Information is identified where it occurs, by an introductory statement to a chapter or section.

Trademarks and Service Marks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

IBM
OS/390
System/390
RACF
CICS/ESA
Language Environment
SAA
MVS/ESA
CICS
BookManager
IMS
OS/2
CICS/MVS
Enterprise Systems Architecture/390
AD/Cycle

Summary of Changes

Changes to the first edition

The following changes have been made to the first edition:

- APARs

PQ04235
Correction of statement that token is converted to uppercase.

PQ05765
Changed wording of error message DFHWB0002 error code 9F21 and added error code 9F21 to message DFHWB1005

PQ06710
Changed abend code in message DFHWB0550 from AWBA to AWB1

PQ07604
Added wbtl_template_abstime and changed order of html_buffer and symbol_list parameters

PQ07658
Changed sample program name ″SIGNON″ to ″DFHWBSN″

PQ08695
Added error codes 9F28 to 9F2D to message DFHWB0002

PQ08889
Responses may be longer than 32k now

PQ08890
Added ″unescape″ function and extra parameters to the interface

PQ11796
Various changes concerning TCP/IP Server Names

PQ15131
DFHWBPA not translating plus signs properly.

PQ15459
Alias transaction fails to start because of insufficient security access.

PQ15555
Reworded the description of wbra_http_header_ptr in the Analyzer reference information

PQ15893
Changed description of decode_input_data_len

PQ16019
In the parser program, we have removed the restriction: “The communication area must not be more that 4096 bytes long.”

PQ16020
wbra_user_data_length parameter is now input and output

PQ16022
DFHWBPA does not return end value padded with values when user data contains multiple values.

PQ18670
Storage growth in the ESDSA where 32K GETMAINS are not being FREEMAINED.

PQ20100
Support provided for user-specified symbol delimiter when calling template manager.

PQ21271
Many DFHWB0002 code (x,9F1E') or DFHWB1005 code (x'9F2E') messages when using the CWI.

PQ21555
AKEC abend in DFHWBWB for large Web messages.

PQ22717
CWI takes a dump if TCP/IP is taken down.

PQ24024
DFHWBCC is not saving ERRNO into GWA when in a SELECT loop during a SEND request.

PQ24105
Different length in WBRA_USER_DATA_LENGTH when using DFHWBUN.

- Minor technical corrections.

Preface

What this book is about

This book describes support in CICS for MVS/ESA 4.1 for the CICS Web Interface. It provides an introduction to the interface, guidance on system and application programming, and reference information. Its contents are as follows:

- "Chapter 1. Introduction" on page 1 describes the facilities provided by the CICS Web Interface.
- "Chapter 2. Setting up the CICS Web Interface" on page 11 describes how to install and set up the CICS Web Interface.
- "Chapter 3. Writing user-replaceable programs for the CICS Web Interface" on page 19 describes how to write user-replaceable programs for the CICS Web Interface.
- "Chapter 4. Writing CICS applications for the CICS Web Interface" on page 41 describes how to write CICS application programs for the CICS Web Interface.
- "Chapter 5. Configuration with the connection manager" on page 55 describes how to configure the CICS Web Interface.
- "Chapter 6. Security" on page 65 describes security considerations for the CICS Web Interface.
- "Chapter 7. Problem determination" on page 69 describes how to diagnose problems with the CICS Web Interface.
- "Appendix A. The business logic interface" on page 85 describes a generalization of the separation of presentation logic from business logic in CICS application design.
- "Appendix B. Messages and Codes" on page 93 lists the messages from the CICS Web Interface.

Who this book is for

This book is written for those responsible for the following tasks:

- Evaluating and planning for the CICS Web Interface. Read "Chapter 1. Introduction" on page 1.
- System administration. Read "Chapter 1. Introduction" on page 1, "Chapter 5. Configuration with the connection manager" on page 55, and "Security for the CICS Web Interface transactions" on page 65.
- Application programming. Read "Chapter 1. Introduction" on page 1, and "Chapter 4. Writing CICS applications for the CICS Web Interface" on page 41.
- System programming. Read "Chapter 1. Introduction" on page 1, "Chapter 2. Setting up the CICS Web Interface" on page 11, "Chapter 3. Writing user-replaceable programs for the CICS Web Interface" on page 19, and "Chapter 6. Security" on page 65.
- Customization. Read "Chapter 1. Introduction" on page 1, and "Chapter 5. Configuration with the connection manager" on page 55.
- Problem determination. Read "Chapter 1. Introduction" on page 1, and "Chapter 7. Problem determination" on page 69.

What you need to know to understand this book

This book assumes that you are familiar with CICS, either as a system administrator or as a system or application programmer, and with the use of web browsers on the World Wide Web.

There are occasional references to details of the hypertext transfer protocol (HTTP) and the hypertext markup language (HTML). As these are only briefly described in "Chapter 1. Introduction" on page 1, you might need to study the references in "HTTP and HTML information" on page xi for more information.

Notes on terminology

When the term "CICS" is used without any qualification in this book, it refers to CICS for MVS/ESA 4.1

"MVS" is used to refer to the Multiple Virtual Storage/Enterprise Systems Architecture (MVS/ESA) operating system.

Determining if a publication is current

IBM regularly updates its publications with new and changed information. When first published, both hardcopy and BookManager softcopy versions of a publication are in step, but subsequent updates will probably be available in softcopy before they are available in hardcopy.

For CICS books, these softcopy updates appear regularly on the *Transaction Processing and Data Collection Kit* CD-ROM, SK2T-0730-xx. Each reissue of the collection kit is indicated by an updated order number suffix (the -xx part). For example, collection kit SK2T-0730-06 is more up-to-date than SK2T-0730-05. The collection kit is also clearly dated on the cover.

Here's how to determine if you are looking at the most current copy of a publication:

- A publication with a higher suffix number is more recent than one with a lower suffix number. For example, the publication with order number SC33-0667-02 is more recent than the publication with order number SC33-0667-01. (Note that suffix numbers are updated as a product moves from release to release, as well as for hardcopy updates within a given release.)
- When the softcopy version of a publication is updated for a new collection kit the order number it shares with the hardcopy version does not change. Also, the date in the edition notice remains that of the original publication. To compare softcopy with hardcopy, and softcopy with softcopy (on two editions of the collection kit, for example), check the last two characters of the publication's file name The higher the number, the more recent the publication. For example, DFHPF104 is more recent than DFHPF103. Next to the publication titles in the CD-ROM booklet and the readme files, asterisks indicate publications that are new or changed.
- Updates to the softcopy are clearly marked by revision codes (usually a "#" character) to the left of the changes.

Bibliography

CICS books for CICS for MVS/ESA

General	
CICS for MVS/ESA Master Index	SC33-1704
CICS for MVS/ESA User's Handbook	SX33-6104
CICS for MVS/ESA Glossary (softcopy only)	GC33-1705
Administration	
CICS for MVS/ESA Installation Guide	SC33-1681
CICS for MVS/ESA System Definition Guide	SC33-1682
CICS for MVS/ESA Customization Guide	SC33-1683
CICS for MVS/ESA Resource Definition Guide	SC33-1684
CICS for MVS/ESA Operations and Utilities Guide	SC33-1685
CICS for MVS/ESA CICS-Supplied Transactions	SC33-1686
Programming	
CICS for MVS/ESA Application Programming Guide	SC33-1687
CICS for MVS/ESA Application Programming Reference	SC33-1688
CICS for MVS/ESA System Programming Reference	SC33-1689
CICS for MVS/ESA Distributed Transaction Programming Guide	SC33-1691
CICS for MVS/ESA Front End Programming Interface User's Guide	SC33-1692
Diagnosis	
CICS for MVS/ESA Problem Determination Guide	GC33-1693
CICS for MVS/ESA Messages and Codes	GC33-1694
CICS for MVS/ESA Diagnosis Reference	LY33-6088
CICS for MVS/ESA Data Areas	LY33-6089
Supplementary Data Areas	LY33-6090
Communication	
CICS for MVS/ESA Intercommunication Guide	SC33-1695
CICS Family: Interproduct Communication	SC33-0824
CICS Family: Communicating from CICS on System/390	SC33-1697
Special topics	
CICS for MVS/ESA Recovery and Restart Guide	SC33-1698
CICS for MVS/ESA Performance Guide	SC33-1699
CICS for MVS/ESA CICS-IMS Database Control Guide	SC33-1700
CICS for MVS/ESA CICS-RACF Security Guide	SC33-1701
CICS for MVS/ESA Shared Data Tables Guide	SC33-1702
CICS for MVS/ESA Transaction Affinities Utility Guide	SC33-1777

Obtaining the books

The *CICS for MVS/ESA Release Guide*, GC33-1570 and the *CICS for MVS/ESA Up and Running*, GC33-1789 contain detailed information about the books in the CICS for MVS/ESA Version 4 Release 1 library. The chapter (the same in both books) discusses both hardcopy and softcopy books and the ways that the books may be ordered. If you have any questions about the CICS for MVS/ESA Version 4 Release 1 library, start with that chapter.

Other CICS books

- *CICS Application Programming Primer (VS COBOL II)*, SC33-0674
- *CICS Application Migration Aid Guide*
- *CICS Family: API Structure*, SC33-1007
- *CICS Family: Client/Server Programming*, SC33-1435
- *CICS Family: General Information*, GC33-0155
- *CICS/ESA Sample Applications Guide* for CICS/ESA Version 4 Release 1, SC33-1173
- *CICS/ESA XRF Guide* for CICS/ESA Version 3 Release 3, SC33-0661

HTTP and HTML information

Information about the hypertext transfer protocol (HTTP) and the hypertext markup language (HTML) is to be found on the World Wide Web.

HTTP

For HTTP information, consult the following:

- *Overview of HTTP*
  `http://www.w3.org/hypertext/WWW/Protocols/Overview.html`
- *Hypertext Transfer Protocol (HTTP/1.0)*
  `http://ds.internic.net/rfc/rfc1945.txt`

The following references are to information about the ISO 8859-1 (Latin-1) character set:

- *ISO 8859-1 National Character Set FAQ*
  http://aliga.cesca.es:1025/%7Ezopcgp01/manuals/ISO8859-1.faq
- *ISO 8859-1:1987* (ordering information)
  http://www.iso.ch/cate/d16338.html
- *ISO 8859-1 (Latin-1) Characters List*
  http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/iso_table.html
- Table of Latin-1 character glyphs
  http://www.w3.org/pub/WWW/MarkUp/Wilbur/latin1.gif

HTML

For HTML information, consult the following:

- *Hypertext Markup Language (HTML)*
  http://www.w3.org/pub/WWW/MarkUp/
- *HTML reference manual*
  http://www.sandia.gov/sci_compute/html_ref.html
- *Hypertext Markup Language - 2.0*
  http://ds.internic.net/rfc/rfc1866.txt
- *HTML, the complete guide*
  http://www.emerson.emory.edu/services/html/html.html
- *The Almost Complete HTML Reference*
  http://www.digital-planet.com/htmlref/contents.html
- *Introducing HTML 3.2*
  http://www.w3.org/pub/WWW/MarkUp/Wilbur/

TCP/IP for MVS

The TCP/IP for MVS Version 3 Release 1 library is as follows:

- *TCP/IP for MVS, VM, OS/2, DOS: Introduction*, GC31-6080
- *IBM TCP/IP for MVS: CICS TCP/IP Socket Interface Guide and Reference*, SC31-7131
- *IBM TCP/IP for MVS: Offloading TCP/IP Processing*, SC31-7133
- *IBM TCP/IP for MVS: Customization and Administration Guide*, SC31-7134
- *IBM TCP/IP for MVS: Programmer's Reference*, SC31-7135
- *IBM TCP/IP for MVS: User's Guide*, SC31-7136
- *IBM TCP/IP for MVS: Application Programming Interface Reference*, SC31-7187
- *IBM TCP/IP: Performance Tuning Guide*, SC31-7188
- *IBM TCP/IP for MVS: Planning and Migration Guide*, SC31-7189
- *IBM TCP/IP for MVS: Quick Reference*, SX75-0095
- *IBM TCP/IP for MVS: Diagnosis Guide*, LY43-0105

Programming

The following manuals might be useful for writing user-replaceable programs for the CICS Web Interface.

C/370

- *C/370 General Information*, GC09-1358
- *C/370 User's Guide*, SC09-1264
- *C/370 Programming Guide*, SC09-1384
- *C/370 Reference and Summary*, SX09-1247

COBOL

- *IBM VS COBOL II Application Programming Guide*, SC26-4045
- *VS COBOL II Application Programming: Language Reference*, SC26-4047
- *VS COBOL II Usage in an CICS/ESA and CICS/MVS Environment*, GG24-3509
- *VS COBOL II Application Programming and Debugging*, SC26-4049

PL/I

- *PL/I Programming: Language Reference*, SC26-4308
- *PL/I Optimizing Compiler Programmer's Guide*, SC33-0006

Assembler

- *Enterprise Systems Architecture/390 Principles of Operation*, SA22-7201
- *High Level Assembler/MVS & VM Programmer's Guide V1R2*, SC26-4941
- *High Level Assembler/MVS & VM Language Reference V1R2*, SC26-4940

Language Environment/370

- *IBM SAA AD/Cycle Language Environment/370 Programming Guide*, SC26-4818
- *IBM SAA AD/Cycle Language Environment/370 Programming Reference*, SC26-3312
- *IBM SAA AD/Cycle Language Environment/370 Planning for Installation and Customization Guide*, SC26-4817

Miscellaneous

The following publications contain related information:

- *Accessing CICS Business Applications from the World Wide Web*, SG24-4547
- *IBM Internet Connection Server for MVS/ESA Up and Running!*, SC31-8204

- *How to Secure the Internet Connection Server for MVS/ESA*, SG324-4803
- *OS/390 Internet BonusPak*, G221-9001
- *IBM's Official Guide to Building a Better Web Site*, SR23-7270
- *CA80: CICS for MVS/ESA EXCI CGI Sample Program*, (none), at URL `http://www.hursley.ibm.com/cics/txppacs/ca80.html`

Chapter 1. Introduction

This chapter is organized as follows:

- "CICS and the Internet" describes the relationship between CICS for MVS/ESA, the CICS Web Interface, TCP/IP for MVS, and web browsers on a network.
- "HTTP requests and responses" on page 3 describes the nature of HTTP requests and responses, and how the CICS Web Interface relates an HTTP request to a CICS program.
- "CICS Web Interface transactions" on page 4 describes the CICS transactions supplied as part of the CICS Web Interface.
- "The CICS Web Interface data set" on page 5 describes the data set in which the CICS Web Interface keeps configuration information.
- "User-replaceable programs" on page 5 describes the user-replaceable programs that you may choose to supply.
- "Application programming with the CICS Web Interface" on page 6 describes writing CICS application programs to exploit HTTP and HTML.
- "Control flow in an HTTP request" on page 6 describes the flow of control in processing a typical HTTP request.
- "Data flow in an HTTP request" on page 8 describes the data flow for a typical HTTP request.
- "Dealing with non-HTTP requests" on page 9 describes how the CICS Web Interface deals with requests that are not HTTP requests.
- "Implementing the CICS Web Interface" on page 9 describes the decisions and activities needed for implementing the CICS Web Interface.

CICS and the Internet

The CICS Web Interface allows web browsers to call programs in a CICS system using the hypertext transfer protocol (HTTP).

Figure 1 shows the relationship between CICS for MVS/ESA, the CICS Web Interface, TCP/IP for MVS, and the web browsers in the network.

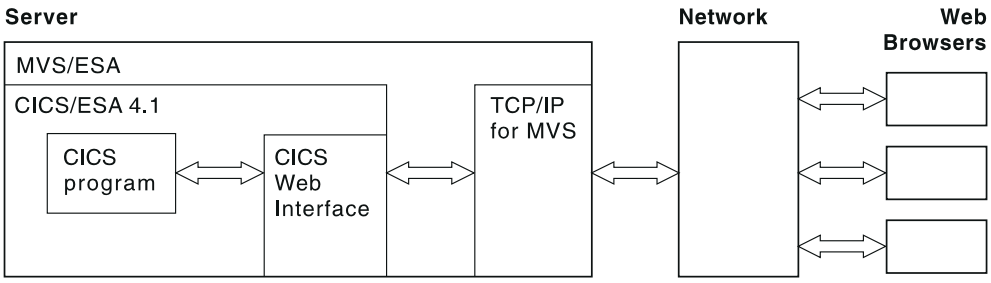

Figure 1. CICS for MVS/ESA and the Internet

A web browser is an HTTP client. The web browser constructs an HTTP request, which is passed across the network to TCP/IP for MVS in the server. TCP/IP for MVS relays the request to the CICS Web Interface, which calls a CICS program to service the request. The output from the CICS program is sent back to the web browser in an HTTP response. The control flow and data flow of this process are

described in more detail in “Control flow in an HTTP request” on page 6 and “Data flow in an HTTP request” on page 8.

The flexible use of CICS programs to supply transaction processing services to different types of requestors depends on a two-tier programming model for CICS applications. In a two-tier design, an application’s *presentation logic* is separated from its *business logic*. The presentation logic is found in such commands as EXEC CICS RECEIVE MAP and EXEC CICS SEND MAP, while the business logic is found in commands like EXEC CICS READ FILE UPDATE, EXEC CICS REWRITE FILE, and in the commands that modify data in databases. By packaging the presentation logic and business logic in separate programs, and using EXEC CICS LINK and a communication area interface to the business logic, the business logic can be made available to a wide range of requestors. “Appendix A. The business logic interface” on page 85 describes the two-tier model, and discusses ways of generalizing access to business logic in CICS applications. In the CICS Web Interface, the presentation logic is packaged in user-replaceable programs called converters. Their functions are described in more detail in “User-replaceable programs” on page 5, and elsewhere in this book.

The CICS Web Interface is one of several methods by which web browsers can have access to CICS transaction processing services. The following brief comparisons are adapted from the information in *Accessing CICS Business Applications from the World Wide Web*. If you use the CICS Internet Gateway, access is by way of the CICS family external presentation interface (EPI). The CICS programs that provide the transaction processing services use a 3270 data stream interface, not a communication area interface. If you use the IBM Internet Connection Server, you provide a common gateway interface (CGI) script that uses the CICS EXCI interface to call CICS programs. In this case the interface to the CICS program is a communication area interface, but the presentation logic has to be put in the CGI script. The IBM Internet Connection Server can use encrypted flows between the web browser and the host, so user IDs and passwords can be securely transmitted over a public network, making it suitable for use on the Internet. The CICS Web Interface does not use encrypted flows, so it is more suitable for a privately-owned communication network, providing transaction processing services from an enterprise’s *intranet*. If you use a secure Internet server on MVS that acts as a secure proxy gateway, you can use the CICS Web Interface securely over public networks.

The CICS Web Interface can be used

- To allow web browsers to use existing CICS programs (for example background tasks that have no principal facility) and the transaction processing services they provide
- To allow web browsers to use newly-created CICS programs that exploit the facilities of HTTP, and of the hypertext markup language (HTML).

The HTTP request is subject to the limitations of the remote procedure call model of distributed computing:

- It is not possible to coordinate changes to recoverable resources in successive requests to the same CICS system.
- Committing changes to recoverable resources is under the control of the CICS system, not the web browser.

The called program executes under a CICS transaction that has no principal facility. It is therefore not allowed to use some commands of the CICS application programming interface:

- Terminal control commands that refer to the principal facility
- Options of EXEC CICS ASSIGN that return terminal attributes
- BMS commands
- Sign-on and sign-off commands

HTTP requests and responses

This section gives a brief outline of the formats of HTTP requests and responses. Detailed information is to be found in the references in "HTTP and HTML information" on page xi. If your primary interest is not programming for the CICS Web Interface, you can omit this section.

A web resource is identified by a uniform resource locator (URL), which identifies the host, and the resource requested. A user of a web browser can enter a URL like the following:

```
http://www.ibm.com:80/Scripts/Global/nph-cc?cc=at
```

In this URL,

- `www.ibm.com` is the name of the host to which the request is to be sent.
- `80` is the TCP/IP port to which the request is to be sent. (80 is the default port for HTTP, and is not usually specified.)
- `/Scripts/Global/nph-cc` is absolute path, identifying a file to be retrieved, or a CGI script to be executed.
- `cc=at` is the query string

The URL is converted by the browser into an HTTP request. An HTTP request consists of a number of HTTP headers followed by optional user data. The headers are separated by the ASCII characters for carriage return and line feed (CRLF). The last header is separated from the user data by a null header, that is two CRLF sequences. The first HTTP header in the request derived from the sample URL above contains:

```
GET /Scripts/Global/nph-cc?cc=at HTTP/1.0
```

The first part of the header is the method (`GET`), the second part is the absolute path and query string, and the last part is the HTTP version.There may be other headers generated by the web browser that sends the request. This request contains no user data.

A common way of generating HTTP requests is by the use of HTML forms. The designer of an HTML form can specify that some of the data entered by the end user is to be transmitted as user data in the HTTP request.A request generated from a form might therefore include user data as well as the headers described above.

In the CICS Web Interface, the HTTP request is received from TCP/IP for MVS, and presented to the analyzer, which is a user-replaceable program. The purpose of the analyzer is to decide what CICS resources are needed to satisfy the request. The interpretation of the absolute path as a file reference is not appropriate in the CICS Web Interface environment, so an enterprise can choose to fix what absolute paths can be sent by browsers, and how the resulting request is interpreted as a request for CICS resources. The functions of the analyzer are described briefly in "The analyzer" on page 5, and in detail in "Writing an analyzer" on page 19. The default analyzer, the URLs that it accepts, and the way it interprets them, are described in "The default analyzer" on page 21.

The HTTP response sent back to the browser consists of headers and optional user data.As in an HTTP request, the CRLF combination separates the headers, and a null header separates the headers from the user data. A typical response might begin with the four headers shown.

```
HTTP/1.0 200 Document follows
Date: Fri, 05 Jul 1996 14:23:02 GMT
Server: NCSA/1.5
Content-type: text/html
```

In the first header, `HTTP/1.0` is the HTTP version, `200` is the HTTP response code, and `Document follows` is a user-readable comment. (There are several standard 3-digit response codes; 200 is a response that indicates successful completion of the request.)The next three headers are the date header, the server header and the content header. The user data might consist of HTML pages, or might be plain text. (In this case the content header promises HTML.)

CICS Web Interface transactions

Three CICS transactions are provided with the CICS Web Interface:

- Connection manager
- Server controller
- Alias

Connection manager (CWBC)

The connection manager is a CICS-supplied transaction that allows you to enable and disable the CICS Web Interface, and to configure and enquire upon it. The use of the connection manager is described in detail in “Chapter 5. Configuration with the connection manager” on page 55. Several instances of this transaction can be running in the CICS system at the same time.

Server controller (CWBM)

The server controller is a CICS-supplied transaction that monitors the TCP/IP for MVS interface for incoming requests. It uses EXEC CICS START to start instances of the alias transaction to service the requests. The server controller is a long-running transaction that is started by the connection manager when the CICS Web Interface is to be enabled, and stopped by it when the CICS Web Interface is to be disabled. Only one instance of the server controller can be running in a CICS system.

Alias (CWBA)

An alias transaction is a CICS-supplied transaction that is started by the server controller to process a single request. Many instances of the alias transaction can be active in a CICS system at the same time, each processing a different request. The alias transaction runs the CICS-supplied alias program that uses EXEC CICS LINK to pass control to the CICS program, as shown in Figure 2 on page 5. If you wish, you may set up additional transaction definitions for alias transactions, each using the CICS-supplied alias program. The use of user-defined alias transactions is described in “Transaction definitions for extra alias transactions” on page 13.

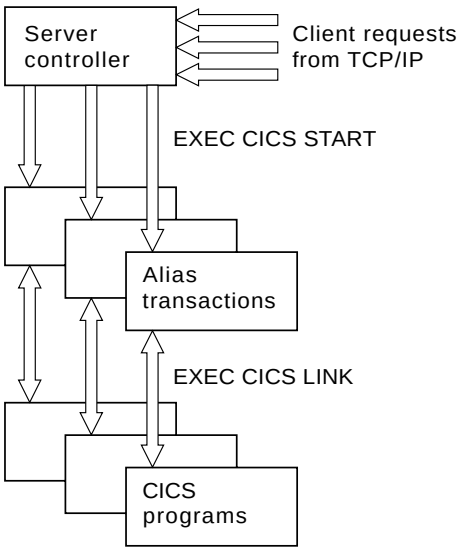

Figure 2. Server controller and alias transactions

The CICS Web Interface data set

The CICS Web Interface uses a data set to hold configuration information. The configuration information is maintained by the operator using the connection manager. For more details about configuration information and the CICS Web Interface data set see “Chapter 5. Configuration with the connection manager” on page 55.

User-replaceable programs

Each incoming request is serviced by a CICS program that provides transaction processing services, and by two other user-replaceable programs, an analyzer (required) and a converter (optional).

The analyzer

There is a single analyzer for the CICS Web Interface in a CICS system. The analyzer is expected to use information in the incoming request to decide what CICS resources are needed to process the request. It can specify:

- The name of the CICS program that is to process the request.
- The name of the converter that is to process the request.
- The name of the alias transaction that is to process the request.
- A user ID or a terminal ID to be associated with the alias transaction.
- Any code page conversion that is needed for user data.
- A modified value for the user data length.

For more information about the analyzer, see “Writing an analyzer” on page 19, and the associated reference material in “Reference information for the analyzer” on page 26.

The converter

You may have many converter programs in a CICS system. Each converter must provide two functions:

- **Decode** is used before the CICS program is called. It can perform the following functions:
  - Use the data from the web browser to build the communication area in the format expected by the CICS program.
  - Supply the lengths of the input and output data in the CICS program communication area.
  - Perform administrative tasks related to the request.
- **Encode** is used after the CICS program has been called. It can perform the following functions:
  - Use the data from the CICS program to build the HTML response or the HTTP response header.
  - Perform administrative tasks related to the response.

For more information about the converter, see "Writing a converter—general" on page 23, "Writing a converter—Decode" on page 23, "Writing a converter—Encode" on page 25, and the associated reference material in "Reference information for the converter" on page 32.

Application programming with the CICS Web Interface

The CICS Web Interface helps you to write CICS application programs that accept HTTP requests as input, and produce HTTP responses as output. The facilities provided are as follows:

- Parse user data in a received HTML form using a parser
- Query information in the HTTP header using an environment variables program
- Build HTML pages from HTML templates using an HTML template manager

In addition there are sample programs that show how you might:

- Introduce a sign-on dialogue for user authentication into the CICS Web Interface.
- Build a simple HTTP response containing HTML to be displayed by the web browser.

For more information about writing CICS applications, see "Chapter 4. Writing CICS applications for the CICS Web Interface" on page 41.

Control flow in an HTTP request

Figure 3 on page 7 shows the control flow for a single HTTP request.

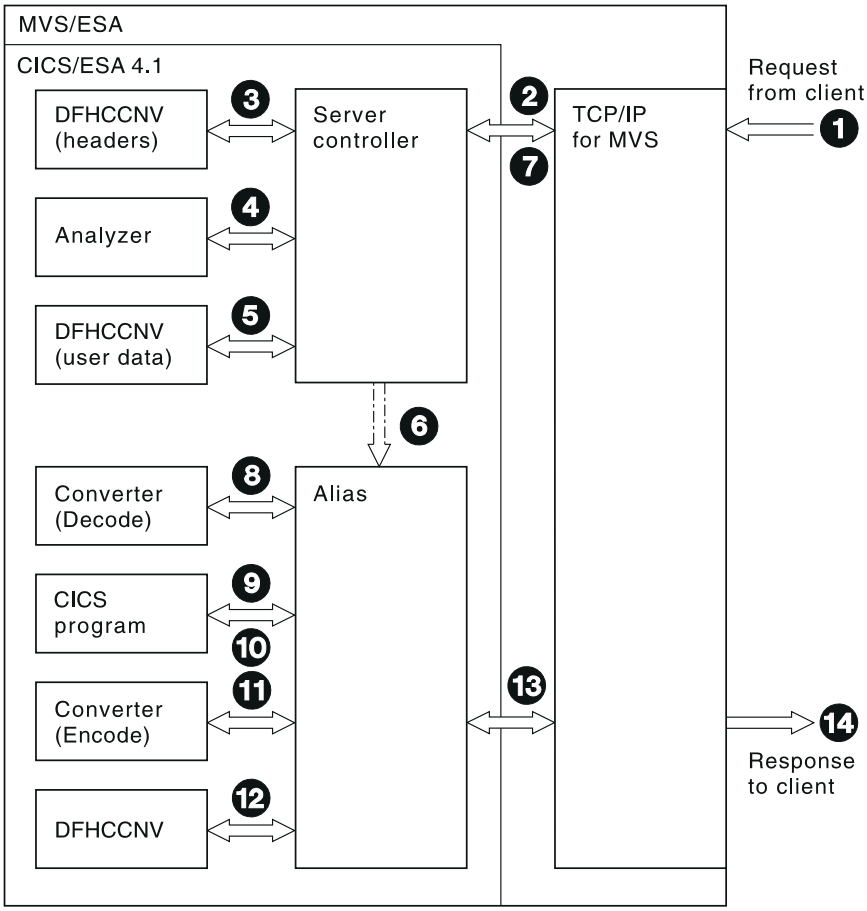

Figure 3. Control flow in an HTTP request

1. An HTTP request arrives in TCP/IP for MVS from a web browser.
2. The server controller monitors the TCP/IP for MVS interface for incoming HTTP requests, and the request is passed to it.
3. The server controller calls DFHCCNV to translate the HTTP request headers from ASCII to EBCDIC. (HTTP headers are always transmitted in ASCII.)
4. The server controller links to the analyzer. The analyzer interprets the request, and specifies the name of the alias transaction and the CICS program to be called to service the request.
5. If the analyzer requests data conversion, the server controller calls DFHCCNV to translate the user data in the HTTP request from the client code page to EBCDIC. (HTTP user data is transmitted in the client code page.)
6. The server controller starts an alias transaction to deal with all further processing of the request in CICS.
7. The server controller returns to monitor the TCP/IP for MVS interface for HTTP requests.
8. If the analyzer requests a converter, the alias calls it, requesting the **Decode** function. **Decode** sets up the communication area for the CICS program.
9. The alias uses EXEC CICS LINK to call the CICS program specified by the analyzer. The communication area passed to the CICS program is the one set up by **Decode**. If no converter program was called, the communication area contains the entire request.

10. The CICS program processes the request, and returns output in the communication area.
11. If the analyzer requested a converter, the alias calls the **Encode** function of the converter, which uses the communication area to prepare the HTTP response. If no converter program was called, the alias assumes that the CICS program has put the desired HTTP response in the communication area.
12. If the analyzer requested data conversion, the alias calls DFHCCNV to translate the HTTP response.
13. The alias returns the results to TCP/IP for MVS, and ends.
14. The response is sent back to the web browser.

Data flow in an HTTP request

Figure 4 shows the data flow from client through CICS and back to the client. As explained in "Control flow in an HTTP request" on page 6, some of these steps are optional.

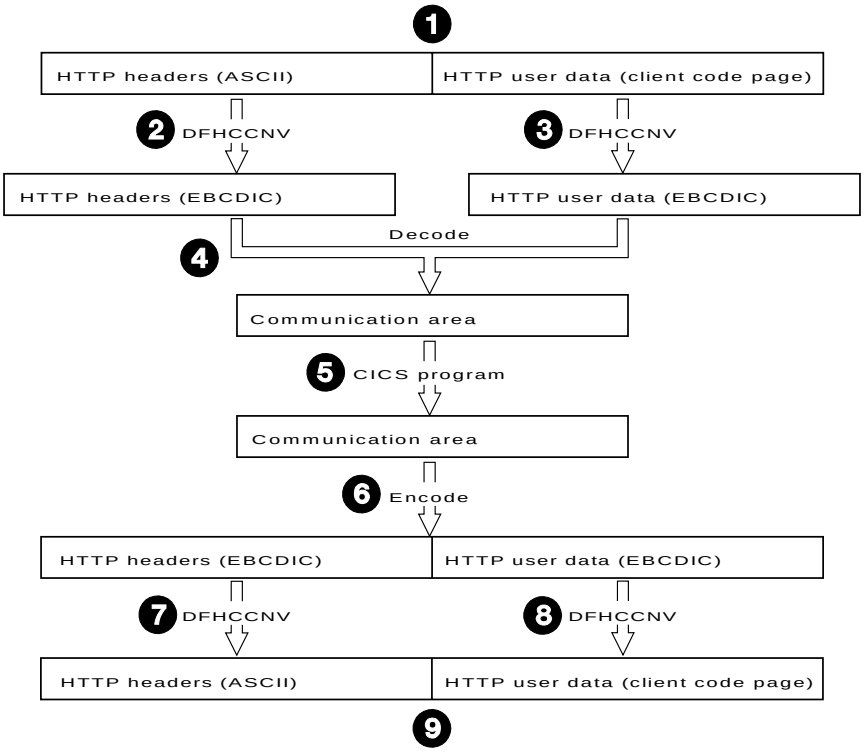

Figure 4. Data flow in an HTTP request

1. A request arrives from a client, and the server controller reads it into a 32K buffer.
2. DFHCCNV translates the HTTP headers from ASCII into EBCDIC.
3. DFHCCNV translates the HTTP user data from the client code page into EBCDIC.
4. The **Decode** function of the converter constructs the communication area for the CICS program. This communication area can be constructed in-place in the buffer provided by the server controller, or **Decode** can free the original buffer and get a new one.
5. The CICS program updates the communication area.

6.

APAR PQ08889
Responses may be longer than 32k.

The **Encode** function of the converter constructs the HTTP response to be sent to the client. The response can be constructed in-place in the communication area, or **Encode** can free the communication area and get a new buffer for the response. The response consists of headers and user data. You can make your response longer than 32K, as described in "Writing CICS programs to process HTTP requests" on page 41.

7. DFHCCNV translates the headers from EBCDIC to ASCII.
8. DFHCCNV translates the user data from EBCDIC to the client code page.
9. The response is sent to the client, and the storage it occupies is freed by the alias.

Dealing with non-HTTP requests

The CICS Web Interface can be used to process requests that are not in the HTTP format. If the server controller cannot parse the incoming request as an HTTP request, the process illustrated in Figure 3 on page 7 is modified in various ways:

- There is no translation of any part of the request before it is passed to the analyzer. The analyzer must do its own translation, or work in the client code page.
- If the analyzer asks for data conversion, the whole of the data is translated before the alias is started.
- If the analyzer detects that there is more data to receive from TCP/IP, it can signal the server controller to get more data.

Implementing the CICS Web Interface

The following list outlines the main steps in preparing to use the CICS Web Interface. Although the steps are presented in order, the principal dependency is that the decisions in steps 1 to 4 should be completed before the activities in steps 5 to 7 are undertaken.

1. Decide which CICS transaction processing services are to be made available to web browsers.

   This might be a mixture of existing services and new services specially created for web browser access.

2. Decide which controls are to be applied to user access to the services.

   If you wish to authenticate users by user ID and password, see "The security sample programs" on page 66 for an example of how this might be done.

3. Decide what the user interface to the services is to be.

   Decide the formats of HTTP requests and responses, and what HTML pages are to be provided to make it easy for the user to send requests. Decide what code pages will be used by the workstations that will be making requests with user data.

4. Decide which services are to be provided by existing CICS programs.
5. Write, test, and install an analyzer.

   The analyzer:

- enforces the controls
- accepts approved requests
- requests appropriate CICS resources.

Writing an analyzer is described in "Chapter 3. Writing user-replaceable programs for the CICS Web Interface" on page 19.

6. Write a converter for each existing CICS program.

   A converter **Decode** function:
   - accepts approved requests
   - constructs the input communication area

   A converter **Encode** function:
   - accepts the output communication area
   - constructs the appropriate HTTP response.

   Writing a converter is described in "Chapter 3. Writing user-replaceable programs for the CICS Web Interface" on page 19.

7. Write new CICS programs (and converters, if you choose to use them) to provide the services not provided by existing CICS programs. You are recommended to use a converter to separate the presentation logic from the business logic.

   Writing new CICS programs that do not need converters is described in "Chapter 4. Writing CICS applications for the CICS Web Interface" on page 41.

Chapter 2. Setting up the CICS Web Interface

This chapter is organized as follows:

- "Prerequisites for using the CICS Web Interface" describes the software requirements for using the CICS Web Interface.
- "Installation process" on page 12 describes how to install the CICS Web Interface.
- "Defining resources to CICS" on page 13 describes the CICS resource definitions you need to make.
- "Changes to TCP/IP for MVS" on page 16 describes changes you can make to TCP/IP for MVS.

Prerequisites for using the CICS Web Interface

This section describes the software requirements for using the CICS Web Interface.

MVS/ESA

The following items must be installed on the MVS/ESA system:

- TCP/IP for MVS Version 3.1 or above. Ports belonging to TCP/IP for MVS must be made available for use by the CICS region involved.
- Language Environment/370. This provides the C run-time libraries that are a prerequisite for running the CICS Web Interface.

  Language Environment/370 is a feature of MVS 5.1 and above, and is available at no charge.

To install the CICS Web Interface you must have System Modification Program/Extended (SMP/E) Release 1.8 or later.

CICS for MVS/ESA

The following must be installed on CICS for MVS/ESA 4.1:

- Two PTFs to allow feature message set use: PTF number UN90128 for non-Kanji support, UN90132 for Kanji support.
- A PTF to fix APAR PN89373 related to CICS Web Interface messages.
- A PTF, number UN85725, related to CICS storage use.
- A PTF, number UN70925, related to the use of the RP TCB. (UN70925 also changes DFHIRP.)
- Three PTFs, numbers UN78778, UN83894, and UN84222, and a PTF to fix APAR number PN70223, to allow features to make and format trace entries, and to format dumps.
- A PTF, number UN91942, making changes to EDF.

CICS must be set up for Language Environment/370 support, as described in the *CICS for MVS/ESA System Definition Guide*.

Note: TCP/IP for MVS CICS Sockets is not a prerequisite for the CICS Web Interface.

TCP/IP for MVS

There are no prerequisites for running the CICS Web Interface.

Installation process

The CICS Web Interface cannot be ordered separately from CICS/ESA 4.1. It is shipped automatically with new orders for the product. Current licensees of CICS for MVS/ESA are notified of the availability of the new function by a program reorder form (or country equivalent form), which may be returned directly to IBM Software Manufacturing Solutions.

The CICS Web Interface is installed separately from base CICS for MVS/ESA. It does not affect installation of CICS for MVS/ESA. The CICS Web Interface is installed from tape into the same libraries as base CICS for MVS/ESA code.

During installation, the CICS Web Interface is linked with TCP/IP for MVS and Language Environment/370 code.

The release of Language Environment/370 used during installation must be the same as the release in which the CICS Web Interface will operate.

If you upgrade TCP/IP for MVS or Language Environment/370, you must relink the module DFHWBWB. Jobs DFHIWBD and DFHIWBL are supplied to do this.

Installing the CICS Web Interface with SMP/E

The CICS Web Interface is installed using SMP/E Release 1.8 or later. Before installing the CICS Web Interface, three libraries have to be defined to SMP/E:

- SCEELKED (for Language Environment/370)
- SEZACMTX (for TCP/IP)

These libraries are needed to link part of the CICS Web Interface.

There is one FMID associated with the CICS Web Interface:

- JCI410A for the CICS Web Interface

Detailed installation instructions are given in the Program Directory supplied with the installation tape.

Other installation tasks

Some other tasks have to be performed to complete the installation.

Creating the CICS Web Interface data set

JCL is provided in the job DFHWBIDC to create the CICS Web Interface data set. The job will be found in the target library SDFHINST after the CICS Web Interface has been installed. The data set is defined as a VSAM key-sequenced data set by a DEFINE CLUSTER command like the following:

```
DEFINE CLUSTER (                        -
             NAME ( XXXXXXXX.CICSWEB.RESOURCE ) -
             CYL  ( 1 1 )               -
             KEYS ( 8 0 )               -
             INDEXED                    -
             VOLUME (vvvvvv)            -
             RECORDSIZE ( 150 150 )     -
             FREESPACE ( 5 5 )          -
             SHAREOPTIONS ( 1 )         -
             )
```

The job to define the data set must be run before you start the connection manager for the first time.

Updating the CICS resource definitions

You must run the DFHWBJ41 job to update the CICS resource definitions with the groups containing the definitions of the CICS Web Interface transactions and the CICS Web Interface data set.

Defining resources to CICS

The CICS Web Interface provides an RDO group defining the CICS resources used by the interface. The following definitions are in the locked group DFHWEB, and cannot be changed:

- Transactions CWBA, CWBC, and CWBM
- Programs supplied with the CICS Web Interface
- Mapsets supplied with the CICS Web Interface

The group DFHWEBF contains the file definition of the CICS Web Interface data set, and you need to change it as described in “Completing the CICS Web Interface data set file definition” on page 14.

The group DFH$WBSN contains the resource definitions for the security sample programs described in “The security sample programs” on page 66.

Other changes that you might need to make are described in the next few sections.

Transaction definitions for extra alias transactions

You may want to use other alias transaction names for various reasons:

- Auditing
- Resource and command checking
- Allocating initiation priorities
- Allocating database plan selection
- Assigning different runaway values to different CICS programs.

If you do want to use other alias transaction names, you must copy the definition of CWBA, making the necessary changes. The definition of CWBA is as follows:

```
DEFINE TRANSACTION(CWBA)    GROUP(DFHWB41)
       PROGRAM(DFHWBA)      TWASIZE(0)
       PROFILE(DFHCICST)    STATUS(ENABLED)
       TASKDATALOC(BELOW)   TASKDATAKEY(USER)
       RUNAWAY(SYSTEM)      SHUTDOWN(ENABLED)
       PRIORITY(1)          TRANCLASS(DFHTCL00)
       DTIMOUT(NO)          INDOUBT(BACKOUT)
       SPURGE(YES)          TPURGE(NO)
       RESSEC(NO)           CMDSEC(NO)
```

You may not change the program name in this definition. Only the CICS-supplied alias program DFHWBA may be used. All the extra alias transactions must be local transactions.

Program definitions for user-replaceable programs

If you are not using autoinstall for programs, you must define all the user-replaceable programs you use. If you are using autoinstall for programs, you do not need to define the converters. In any case analyzers must be defined with

EXECKEY(CICS). All the user replaceable programs must be local to the system in which the CICS Web Interface is operating.

Transient data definitions

You must supply DCT definitions for the CICS Web Interface message transient data queue. The following sample is supplied with the CICS Web Interface as member DFHWBDCT of the SDFHSAMP target library:

```
DFHDCT TYPE=SDSCI,
       BLKSIZE=137,
       RECSIZE=133,
       DSCNAME=CWBO,
       RECFORM=VARUNB,
       BUFNO=1,
       TYPEFLE=OUTPUT

DFHDCT TYPE=EXTRA,
       DESTID=CWBO,
       DSCNAME=CWBO
```

If you define the destination in the manner of the sample, you must also add a suitable DD statement for the extrapartition queue to the CICS JCL, for instance:

```
//CWBO DD SYSOUT=A
```

The destination could also be made intrapartition or indirect.

Completing the CICS Web Interface data set file definition

The file definition for the CICS Web Interface data set is supplied as follows, and it needs to be modified before you enable the CICS Web Interface.

```
DEFINE FILE(DFHWBCD)          GROUP(DFHWEBF)
       DSNAME(xxxxxxxx.CICSWEB.RESOURCE)
       LSRPOOLID(1)           STRINGS(1)
       STATUS(ENABLED)        OPENTIME(FIRSTREF)
       DATABUFFERS(2)         INDEXBUFFERS(1)
       RECORDFORMAT(F)
       ADD(YES)               BROWSE(YES)            DELETE(YES)
       READ(YES)              UPDATE(YES)
       DESCRIPTION(CICS Web Interface data set definition)
```

You use CEDA to update the definition to supply an appropriate name for the VSAM data set. DFHWEBF is a locked group, so you must copy the file definition to your own group and change it there.

The VSAM definition for this file is in member DFHWBIDC of SDFHINST, and is as follows:

```
DEFINE CLUSTER (                        -
             NAME ( XXXXXXXX.CICSWEB.RESOURCE ) -
             CYL  ( 1 1 )               -
             KEYS ( 8 0 )               -
             INDEXED                    -
             VOLUME (vvvvvv)            -
             RECORDSIZE ( 150 150 )     -
             FREESPACE ( 5 5 )          -
             SHAREOPTIONS ( 1 )         -
             )
```

Setting up a PDS for the template manager

If you use the HTML template manager for constructing HTTP responses, you may provide an MVS partitioned data set to hold the templates. You can use ISPF to

create the templates as members of this data set. The record format can be FB (fixed blocked), VB (variable blocked), or U (undefined). The templates can contain sequence numbers as follows:

- VB format: the sequence numbers must be in record positions 1 through 8.
- FB format, and LRECL 80: the sequence numbers must be in record positions 73 through 80.

In any other case, there must be no sequence numbers in the records. The template manager decides whether there are sequence numbers by looking at the first logical record of a member of the PDS, so members that are only partially sequenced might be interpreted incorrectly.

The data set must be defined in a new DD statement, DD name DFHHTML, in the CICS JCL.

Defining a conversion table

You need to create or modify a DFHCNV table for data conversion to allow the server controller to deal with incoming requests. The use of the DFHCNV macro for defining the table is described in *CICS Family: Communicating from CICS on System/390*. There are two kinds of data conversion performed in the CICS Web Interface:

- Conversion of the HTTP header information. This information is always transmitted as ASCII data using the ISO 8859-1 (Latin-1) character set. This is the base character set for HTTP and HTML. This data has to be translated into EBCDIC. The conversion template name that the server controller supplies to the DFHCCNV program, which does the translation, is DFHWBHH.
- Conversion of the HTTP user data. This information is transmitted in the code page of the HTTP client, and can be translated into EBCDIC if required. The conversion template name is supplied by the analyzer. If the request is not an HTTP request, all the request is translated using the name supplied by the analyzer.

For data conversion of the HTTP headers, you need to create a conversion template as follows:

```
DFHCNV TYPE=ENTRY,                                          *
      RTYPE=PC,                                             *
      CLINTCP=437,                                          *
      SRVERCP=037,                                          *
      RNAME=DFHWBHH,                                        *
      USREXIT=NO
DFHCNV TYPE=SELECT,OPTION=DEFAULT
DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=32767,   *
      LAST=YES
```

In the TYPE=ENTRY macro, the RNAME parameter must be DFHWBHH. The code page specifications CLINTCP and SRVERCP will get the HTTP request headers translated from ASCII to EBCDIC, and the HTTP response headers translated from EBCDIC to ASCII. The TYPE=SELECT and TYPE=FIELD macros must be coded exactly as shown.

For each name that the analyzer might specify for translating user data in the request from the client code page into EBCDIC, and for translating the user data in the response from EBCDIC to the client code page, you need to create a conversion template as follows:

```
DFHCNV TYPE=ENTRY,                                          *
      RTYPE=PC,                                             *
      CLINTCP=437,                                          *
      SRVERCP=037,                                          *
      RNAME=DFHWBUD,                                        *
      USREXIT=NO
DFHCNV TYPE=SELECT,OPTION=DEFAULT
DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=32767,   *
      LAST=YES
```

In the TYPE=ENTRY macro, the CLINTCP parameter must specify the code page of the client, and the RNAME parameter must specify the name that the analyzer will supply. The sample entry above supports translation of user data in the request from ASCII to EBCDIC, and of the user data in the response from EBCDIC to ASCII, for the default analyzer, which uses the name DFHWBUD. You may code the TYPE=SELECT and TYPE=FIELD macros in any way that is appropriate to the format of the user data that the client sends.

You may use the TYPE=INITIAL macro to set defaults for some of the values specified in these samples, as explained in *CICS Family: Communicating from CICS on System/390*.

The following sample shows a complete definition of the conversion templates for use with a web browser using a Japanese double-byte character set. The code page 932 is one of several code pages for Japanese web browsers, and 931 is one of the corresponding System/390 code pages. This sample can be used with the default analyzer.

```
DFHCNV TYPE=INITIAL
DFHCNV TYPE=ENTRY,RTYPE=PC,RNAME=DFHWBHH,USREXIT=NO,          *
      SRVERCP=037,CLINTCP=437
DFHCNV TYPE=SELECT,OPTION=DEFAULT
DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=32767,   *
      LAST=YES
DFHCNV TYPE=ENTRY,RTYPE=PC,RNAME=DFHWBUD,USREXIT=NO,          *
      CLINTCP=932,SRVERCP=931
DFHCNV TYPE=SELECT,OPTION=DEFAULT
DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=32767,   *
      LAST=YES
DFHCNV TYPE=FINAL
END
```

Running the sample application

A sample application DFH$WB1A is provided to help you test the operation of the CICS Web Interface. Use the connection manager to enable the interface with the default analyzer, as described in “Chapter 5. Configuration with the connection manager” on page 55. From a suitable web browser, enter a URL that connects to the CICS Web Interface with absolute path `/CICS/CWBA/DFH$WB1A`. The response displays the message “`DFH$WB1A on system XXXXXXXX successfully invoked through the CICS Web Interface.`” with `XXXXXXXX` replaced by the application ID of the CICS system in which the CICS Web Interface is running.

Changes to TCP/IP for MVS

If you wish to reserve the port on which the CICS Web Interface listens for incoming client requests, you can specify the PORT option in the tcpip.PROFILE.TCPIP data set, as described in *IBM TCP/IP for MVS: Customization and Administration Guide*.

You are recommended to reserve a port for the CICS Web Interface, and to ensure that the CICS Web Interface has exclusive use of that port.

The maximum length of any queue of requests for a TCP/IP port on which a program is listening is controlled by the SOMAXCONN parameter in the tcpip.PROFILE.TCPIP data set. The server controller is a program that listens on a TCP/IP port, so you must coordinate the value of this parameter with the value chosen for the Backlog panel field in the connection manager panels DFHWB02, and DFHWB12.

The CICS Web Interface needs to access a name server during its operation. You must specify the address of the name server in a file allocated to the SYSTCPD DD statement in your CICS JCL. The contents of this file are described in *IBM TCP/IP for MVS: Customization and Administration Guide*. You must specify at least the following:

```
NSINTERADDR n.n.n.n
```

where `n.n.n.n` is the dotted decimal address of the name server.

APARs PQ11796 and PQ16996

Additional information

If you do not provide a name server address:

- The security sample program DFH$WBSN does not execute correctly.
- The environment variables program DFHWBENV does not return the connection name in SERVER_NAME, but the dotted decimal address of the connection, and it also returns a null string for REMOTE_HOST

Chapter 3. Writing user-replaceable programs for the CICS Web Interface

This chapter is organized as follows:

- "Writing the user-replaceable programs" gives guidance about the design and coding of analyzers and converters.
- "Reference information for the analyzer" on page 26 gives detailed information about the parameters for the analyzer.
- "Reference information for the converter" on page 32 gives detailed information about the parameters for the **Decode** and **Encode** converter functions.

This chapter contains Product-sensitive Programming Information and Associated Guidance Information.

Writing the user-replaceable programs

The user-replaceable programs are the analyzer and the converter.

You must supply an analyzer, or use the IBM-supplied default analyzer DFHWBADX.

You might not need to write any converters. If the analyzer indicates that a converter is not required, the entire request is passed to the CICS program in its communication area.

You may write your user-replaceable programs in Assembler, C, COBOL, or PL/I. Language-dependent header files, include files, and copy books are described in the reference sections.

Writing an analyzer

The analyzer interprets the incoming request and specifies the CICS resources that are needed to provide the requested service.

Inputs

The analyzer input includes:

- An eye-catcher for an analyzer parameter list
- The IP address of the client
- The IP address of the server
- An indicator of whether the request is an HTTP request

The analyzer input also includes the incoming request. If the request is an HTTP request, various parts of the request are identified by pointers and lengths to make processing easier:

- Version
- Method
- Absolute path
- Request header
- User data

(The version, method, absolute path, and request header are in EBCDIC, but the user data is in the client code page.)

If the request is not an HTTP request, the input includes the entire request in the client code page.

Outputs

The analyzer must provide the following output:

- A response code

It may also provide the following outputs:

- The name of the CICS program that is to service the request
- The conversion template name for code page translation of the user data
- The transaction ID of the alias transaction that is to service the request
- The name of the converter that is to be used to service the request
- A user token that is to be passed to the converter functions
- The user ID or the terminal ID for EDF for the alias transaction
- A value indicating if the server controller or the parser is to unescape the user data
- A modified value for the user data length
- A reason code

Processing

The inputs and outputs are presented in a CICS communication area. The analyzer can use any of its inputs to determine the CICS resources that are to be used to service the request, and the other outputs it might wish to supply.

If there are rules about which clients can use which services, you must use the input client IP address and the contents of the request to decide if this client is allowed to use this service. You can reject a client request by setting the output response to a value other than URP_OK.

If the code page of the client is not an EBCDIC code page, you must set the output conversion template name. See “Code page considerations” on page 21.

If the CICS program to be used to service the request can be determined by the analyzer, you should set the output CICS program name. (If you do not set it here, you must specify the use of a converter, and the converter **Decode** function must set the program name, or no CICS program will be called.)

If the selected CICS program needs a converter, you must set the output converter name.

If the service is to be provided under a user-defined alias transaction, you must set the output transaction name.

If you wish to pass other information to the converter functions, you should set an output user token. This token could be a pointer to (shared) storage acquired by the analyzer to be freed by the converter. You may also make changes to the contents of the request, and these will be visible to **Decode** if a converter is used, or to the CICS program if no converter is used.

If you want to use EDF to test your CICS programs or converters, you should specify an output terminal ID. The use of EDF is described in “Using EDF” on page 82.

You can use various return codes and reason codes to report errors in the inputs and processing. If the request is an HTTP request, some of the responses are associated with architected HTTP responses. For details consult “Reference information for the analyzer” on page 26. If you use any response other than

URP_OK, or if you use any reason codes, you should document the responses and reason codes to help with problem determination.

If the request is a non-HTTP request, and you detect that there is more data to be received, you can use the URP_EXCEPTION response and the URP_RECEIVE_OUTSTANDING reason code to get the server controller to receive more data, and add it to that already in the input area. The server controller then calls the analyzer again.

The analyzer must tell CICS whether the HTTP user data is to be treated as escaped or unescaped during the lifetime of the alias transaction, and whether the server controller is to unescape the data on return from the analyzer. The values returned affect the decision to unescape made by the parser (DFHWBPA) when run in the converter or the application code. The values available for the wbra_unescape parameter are given in "Reference information for the analyzer" on page 26.

Code page considerations

When designing your analyzer, if you are not using the HTML base code page ISO 8859-1 (Latin-1) for user data, you need to specify the conversion template for the code pages used. You must perform the following steps:

1. Identify the character sets that HTTP clients will be using. All the browsers that have access to the CICS Web Interface might use the same code page, or you might be able to tell the code page from the IP address of the client. It might be possible to get the browsers to create URLs that include an indicator of the code page. The HTTP request headers Content-Type and Content-Language might contain useful information, but they are not used consistently by all web browsers.
2. Use *CICS Family: Communicating from CICS on System/390* to decide the kind of conversion to be performed, and add a conversion template to the DFHCNV table. For nonstandard conversion you need to create or modify the DFHUCNV program.
3. Write an analyzer that decides what data conversion is needed, and sets the name of the conversion template in **wbra_dfhcnv_key**.

Performance considerations

The analyzer is called for each HTTP request that is received. Each inbound request must be processed by the analyzer before the server controller can accept the next request from TCP/IP for MVS, so delays in the analyzer can affect the overall throughput of the system.

You should use performance-efficient techniques such as index tables to resolve the relations between request and CICS resources, rather than performing I/O operations. You should avoid allocating or freeing storage, since this can introduce processing delays.

The default analyzer

A default analyzer DFHWBADX is supplied with the CICS Web Interface. The source code for the analyzer is supplied in various languages, and you can use it as the basis of your own analyzer. The source files are as follows:

- DFHWBADX (Assembler)
- DFHWBAHX (C)
- DFHWBALX (PL/I)
- DFHWBAOX (COBOL)

The default analyzer is written for HTTP requests in which the absolute path has one of the following forms:

/converter/alias/program?token
/converter/alias/program
/converter/alias/program/filename
/converter/alias/program/filename?token
/converter/alias/program/?token

The default analyzer checks the eye-catcher, and then interprets the contents of the absolute path as follows:

- *converter* must be between 1 and 8 characters long. It is converted to uppercase and interpreted as the name of the converter to be called by the alias, unless it has the value "CICS", in which case the converter name is set to nulls to show that no converter is to be used.
- *alias* must be between 1 and 4 characters long. It is converted to uppercase and interpreted as the transaction ID of the alias transaction to be used to service the request.
- *program* must be between 1 and 8 characters long. It is converted to uppercase and interpreted as the name of the CICS program that is to be used to service the request.
- *filename* can be any length, but it must not begin with a slash ("/") or contain a question mark. It must be made up of characters allowed in URLs. It is ignored by the analyzer, but is available to the converter or the CICS program.
- *token*, if it is present, must be between 1 and 8 characters long. It is interpreted as the user token to be passed to the converter.

The default analyzer sets the conversion template name to DFHWBUD.

The default analyzer diagnoses various errors, and the meanings of its responses and reason codes are described in "DFHWBADX responses and reason codes" on page 32.

APAR PQ08890
Added ″unescape″

The default analyzer checks if unescape=no has been specified. If so, the analyzer links to unescaping module DFHWBUN to unescape any user data.

Reference Information for the Unescaping Program

The CICS Web Interface Unescaping program is DFHWBUN. It parses data in a commarea replacing escaped characters in the format %nn with their unescaped equivalent. It also changes the ascii ″+″ characters('2B'x) in the HTTP input to ascii space characters ('20'x). The program is called by EXEC CICS LINK with a commarea in the following format:

Parameters for the unescaping module			
Offset	Length	Type	Notes
0	4	F	Address of data to be unescaped. Required input for each call.
4	2	X	Length of data to be unescaped. Required input for each call.

6	2	X	Length of data after unescaping. Output after every call.
8	4	X	Return Code. Output after every call.

The Return Codes are as follows:

0 - Unescaping was successfully performed.

8 - Error. No unescaping was performed due to a Length field of zero.

Writing a converter—general

The converter provides **Decode** and **Encode** functions for processing a request.

Inputs

The converter input includes:

- An indicator of the function (**Decode** or **Encode**) that is to be performed
- Parameters for the function, as described in later sections

Outputs

The converter output must include a response, and might include a reason code. The outputs are described in more detail for each function.

Processing

The inputs and outputs are presented in a CICS communication area. On entry to the converter, is should check the input field **converter_function** to see whether the requested function is **Decode** or **Encode**. The rest of the processing depends on the function requested.

Performance considerations

The converter is called from the alias transaction, and therefore its functions can only affect the performance of a single client request.

You should avoid operations that introduce processing delays. If a converter functions needs to allocate storage, it should use the NOSUSPEND option of EXEC CICS GETMAIN. The efficiency of later processing can be improved if **Decode** sets **decode_input_data_len** to the exact length of the data to be passed to the CICS program, since this optimizes the use of storage and data transmission facilities.

Writing a converter—Decode

This section gives informal descriptions of the inputs and outputs of **Decode**, and gives some hints about processing.

Inputs

The inputs to **Decode** include:

- An eye-catcher for a **Decode** parameter list
- The IP address of the client
- The name of the CICS program that is to service the request, if this was set by the analyzer
- A pointer to the buffer containing the request as modified by the analyzer
- The user token supplied by the analyzer

If the incoming request is an HTTP request, various parts of the request are identified by pointers and lengths to make processing easier:

- Version
- Method

- Absolute path
- Request header
- User data

Outputs

Decode must set the following outputs:

- A response code
- The length of the communication area to be passed to the CICS program

It might also provide the following outputs:

- A pointer to the communication area to be passed to the CICS program, if this is not the input communication area.
- The name of the CICS program that is to service the request.
- The user token to be passed to **Encode**
- A reason code

Processing

The main purpose of **Decode** is to provide the communication area for the CICS program.

If your converter is running as part of the CICS Web Interface, or as part of the CICS business logic interface in *pointer* mode, the buffer can occupy the same storage as the communication area returned by the CICS program, or you can use EXEC CICS GETMAIN to get new storage, and EXEC CICS FREEMAIN to free the storage occupied by the input communication area. (That is, if **decode_data_ptr**is altered to address another storage location, it is the **decode** program's responsibility to always FREEMAIN the original storage.) If you do not free the input communication area, storage will be lost. In this case you must set the output pointer to the new communication area.

If your converter is running as part of the CICS business logic interface in *offset* mode, the buffer must occupy the same storage as the input communication area. In this case you must not use EXEC CICS GETMAIN to get new storage, and you must not change the data pointer in the parameter list.

You may set the output for the length of the communication area you pass to the CICS program, and you may set an output for the returned length if this is less than the length to be passed to the CICS program.

You may use the input user token passed by the analyzer, and if this is a pointer, you may use and update the information in the storage it addresses. You may pass the same token on to **Encode**, or you may replace it with another token. If you use EXEC CICS GETMAIN to acquire storage for **Encode**, you must use EXEC CICS FREEMAIN to free the storage that was acquired by the analyzer.

The CICS program name as set by the analyzer is available for your use, and you may change it. If the analyzer did not set the name of the CICS program, you must set it here, or no CICS program will be called.

You can use various return codes and reason codes to report errors in the inputs and processing. If the request is an HTTP request, some of the responses and reason codes are associated with architected HTTP responses. For details consult "Reference information for the converter" on page 32. If you use any response other than URP_OK, or if you use any reason codes, you should document the responses and reason codes to help with problem determination.

Writing a converter—Encode

This section gives informal descriptions of the inputs and outputs of **Encode**, and gives some hints about processing.

Inputs

The inputs to **Encode** include:

- An eye-catcher for an **Encode** parameter list
- A pointer to the communication area returned by the CICS program, and its length
- The user token created by the analyzer and passed by **Decode**

Outputs

Encode must set the following outputs:

- A response code
- A pointer to the buffer containing the response to be sent to the client

It might also provide the following outputs.

- A reason code

Processing

The main purpose of **Encode** is to provide the response to be sent to the client. You can use the HTML template manager to help you to construct the HTTP response; see "Reference information for the HTML template manager" on page 42. You must set the output response length, and you must put the data length (response length plus 4) in the first word of the buffer.

If your converter is running as part of the CICS Web Interface, or as part of the CICS business logic interface in *pointer* mode, the buffer can occupy the same storage as the communication area returned by the CICS program, or you can use EXEC CICS GETMAIN to get new storage, and EXEC CICS FREEMAIN to free the storage occupied by the input communication area. If you do not free the input communication area, storage will be lost. In this case you must set the output pointer to the new communication area. (That is, if **encode_data_ptr**is altered to address another storage location, it is the **encode** program's responsibility to always FREEMAIN the original storage.)

If your converter is running as part of the CICS business logic interface in *offset* mode, the buffer must occupy the same storage as the input communication area. In this case you must not use EXEC CICS GETMAIN to get new storage, and you must not change the data pointer in the parameter list.

You may use the input user token passed by **Decode**, and, if this is a pointer, you may use the information in the storage it addresses. If it is a pointer, you must use EXEC CICS FREEMAIN to free the storage it addresses.

You can use various return codes and reason codes to report errors in the inputs and processing. If the request is an HTTP request, some of the responses and reason codes are associated with architected HTTP responses. For details consult "Reference information for the converter" on page 32. If you use any response other than URP_OK, or if you use any reason codes, you should document the responses and reason codes to help with problem determination.

Reference information for the analyzer

This section provides:

- Reference information for the analyzer
- Information about the responses and reason codes for the default analyzer, DFHWBADX

Analyzer

Summary of parameters

The names of the parameters and constants, translated into appropriate forms for the different programming languages supported, are defined in files supplied as part of the CICS Web Interface. The files for the various languages are listed in the following table.

Table 1. Parameter files and Constants files for different programming languages

Language	Parameters file	Constants file
Assembler	DFHWBTDD	DFHWBUCD
C	DFHWBTDH	DFHWBUCH
COBOL	DFHWBTDO	DFHWBUCO
PL/I	DFHWBTDL	DFHWBUCL

These files give language-specific information about the data types of the fields in the communication area. If you use these files you must specify XOPTS(NOLINKAGE) on the Translator step; failure to do this causes the compile to fail.

In the following table, the names of the parameters are given in abbreviated form: each name in the table must be prefixed with **wbra_** to give the name of the parameter.

APAR PQ08890

Added http_header_ptr and unescape

Table 2. Parameters for the analyzer

Input wbra_	Inout wbra_	Output wbra_
client_ip_address **eyecatcher** **function** **http_header_ptr** **http_version_length** **http_version_ptr** **method_length** **method_ptr** **request_header_length** **request_header_ptr** **request_type** **resource_length** **resource_ptr** **server_ip_address** **user_data_length**	**unescape** **user_data_ptr** **user_token**	**alias_termid** **alias_tranid** **content_length** **converter_program** **dfhcnv_key** **reason** **response** **server_program** **userid**

Function

The analyzer is called by the server controller before it starts the alias. The analyzer can examine the incoming request, and must specify the CICS resources needed to process the request.

Parameters

wbra_alias_termid
(Output only)

A string of length 4. The terminal ID of the terminal at which the transaction is to be started. If you do not set this field, the alias is started without a terminal.

wbra_alias_tranid
(Output only)

A string of length 4. The transaction ID of the alias that is to service the request. If you do not set this field, or if you set it to blanks, CWBA is used.

wbra_client_ip_address
(Input only)

The 32-bit IP address of the client.

wbra_content_length
(Input only)

A 32–bit binary representation of the user data length as specified by the 'Content-Length' HTTP header in the received data.

wbra_converter_program
(Output only)

A string of length 8. The name of the converter whose **Decode** and **Encode** functions are used to process the request. If you do not set this field, no converter is called.

wbra_dfhcnv_key
(Output only)

A string of length 8. The name of the conversion template in the DFHCNV table for the code page translation of the user data for this request. If the request is not an HTTP request, this name is used to translate the entire request. The name you chose must be defined in the DFHCNV table, as described in "Defining a conversion table" on page 15. If you do not set this field, there is no translation.

wbra_eyecatcher
(Input only)

A string of length 8. Its value is "`>analyze`".

wbra_function
(Input only)

A code indicating that the analyzer is being called. The value is 1.

APAR PQ15555
Reworded description of wbra_http_header_ptr

wbra_http_header_ptr
(input only)

A pointer to a copy of the string identifying the HTTP absolute path specified in the HTTP request, which has not been unescaped. The length of the string is specified by **wbra_resource_length**.

wbra_http_version_length
(Input only)

The length in bytes of the string identifying the HTTP version supported by the client. If the request is not an HTTP request, this length is zero.

wbra_http_version_ptr
(Input only)

A pointer to the string identifying the HTTP version supported by the client. If the request is not an HTTP request, do not use this pointer.

wbra_method_length
(Input only)

The length in bytes of the string identifying the method specified in the HTTP request. If the request is not an HTTP request, this length is zero.

wbra_method_ptr
(Input only)

A pointer to the method specified in the HTTP request. If the request is not an HTTP request, do not use this pointer.

wbra_reason
(Output only)

A reason code—see "Responses and reason codes" on page 31. When **wbra_response**= URP_EXCEPTION, reason codes in the range 1000–1999 are reserved by CICS and relate to actions taken by the server controller.

wbra_request_header_length
(Input only)

The length of the first HTTP header in the HTTP request. If the request is not an HTTP request, this length is zero.

wbra_request_header_ptr
(Input only)

A pointer to the first HTTP header in the HTTP request. The other HTTP headers follow this one in the request buffer. If the request is not an HTTP request, do not use this pointer.

wbra_request_type
(Input only)

If this is an HTTP request, the value is WBRA_REQUEST_HTTP. If this is not an HTTP request, the value is WBRA_REQUEST_NON_HTTP.

wbra_resource_length
(Input only)

The length in bytes of the string identifying the HTTP absolute path specified in the HTTP request. If the request is not an HTTP request, this length is zero.

wbra_resource_ptr
(Input only)

A pointer to the string identifying the HTTP absolute path specified in the HTTP request. If the request is not an HTTP request, do not use this pointer.

wbra_response
(Output only)

A response—see “Responses and reason codes” on page 31.

wbra_server_ip_address
(Input only)

The 32-bit IP address of the TCP/IP for MVS region receiving the request.

wbra_server_program
(Output only)

A string of length 8. The name that is passed to **Decode** as **decode_server_program**. If you do not set this field, the value passed is nulls. The program name must be set here or in the **Decode** function of the converter specified in **wbra_converter_program**, or no CICS program will be called.

wbra_unescape
(InOut)

On input this field is set to reflect the current setting of the Unescape option of the Web Interface (see Figure 7 on page 60), that is, it is set to **wbra_unescape_required** if the Unescape option is set to Yes, otherwise it is set to **wbra_unescape_not_required**. On output, if this field is set to **wbra_unescape_required**, the HTTP user data is assumed to be in an escaped form, and CICS performs unescaping on return from the analyzer. If this field is set to **wbra_unescape_completed**, the HTTP user data is assumed to be in an unescaped form and CICS does not perform unescaping. If this field is set to **wbra_unescape_not_required**, the HTTP data is assumed to be in an escaped form and CICS does not perform unescaping. This value is ignored if the request is not an HTTP request.

This field also determines if the parser (DFHWBPA) is to unescape the value it returns when run in either the converter or the application code. If this field is set to **wbra_unescape_not_required**, the parser unescapes the value it returns. If this field is set to any other value, the parser does not perform unescaping. See “Reference information for the parser program” on page 53 for details of the unescaping provided by the parser.

wbra_user_data_length
(Input and output)

A 15–bit binary integer representing the length of the user data in the HTTP request. If the request is not an HTTP request, this length is the length of the request. The length passed to the analyzer includes any trailing carriage return and line feed (CRLF) characters which may delimit the end of the user data. If the length is reduced, the newly redundant bytes are replaced by null characters. The modified value is passed on to the business logic interface program in parameter wba1_user_data_length, and to the decode program in decode_user_data_length.

wbra_user_data_ptr
(Input only)

A pointer to the user data in the HTTP request. If the request is not an HTTP request, this is a pointer to the request.

wbra_user_token
(Output only)

A 64-bit token that is passed to **Decode** as **decode_user_token**. If you do not set this field, the value passed is null. If there is no converter for this request, the value is ignored.

wbra_userid

(Output only)

A string of length 8. If you do not specify **wbra_alias_termid**, this field is the user ID for the alias. If you specify **wbra_alias_termid**, this field is ignored. If you specify neither, the alias runs under the same user ID as the server controller.

Responses and reason codes

You must return one of the following values in **wbra_response**:

URP_OK

The server controller starts the alias transaction.

URP_EXCEPTION

The alias transaction is not started. The server controller writes an exception trace entry (trace point A00F), and issues a message (DFHWB0523).

If the request is an HTTP request, response 400 is sent to the Web browser.

If the request is not an HTTP request, and the reason code is not URP_RECEIVE_OUTSTANDING, no response is sent, and the TCP/IP for MVS socket is closed.

If the request is not an HTTP request, and the reason code is URP_RECEIVE_OUTSTANDING, the server controller receives more data from TCP/IP for MVS, and adds it to that already in the input area. It calls the analyzer again.

URP_INVALID

The alias transaction is not started. The server controller writes an exception trace entry (trace point A00F), and issues a message (DFHWB0523). If the request is an HTTP request, response 400 is sent to the web browser. If the request is not an HTTP request, no response is sent, and the TCP/IP for MVS socket is closed.

URP_DISASTER

The alias transaction is not started. The server controller writes an exception trace entry (trace point A00F), and issues a message (DFHWB0523). If the request is an HTTP request, response 400 is sent to the web browser. If the request is not an HTTP request, no response is sent, and the TCP/IP for MVS socket is closed.

If you return any other value in **wbra_response**, the server controller writes an exception trace entry (trace point A00F), and issues a message (DFHWB0523). If the request is an HTTP request, response 400 is sent to the web browser. If the request is not an HTTP request, no response is sent, and the TCP/IP for MVS socket is closed.

You may supply a 32-bit reason code in conjunction with the response value to provide further information in error cases. The CICS Web Interface does not take any action on the reason code returned by the analyzer. The reason code is output in any trace entry that results from the invocation of the analyzer, and in message DFHWB0523.

DFHWBADX responses and reason codes

The meanings of the responses produced by the default analyzer DFHWBADX are as follows:

URP_OK
The analyzer found that the request conformed to the default HTTP request format, and generated the appropriate outputs for the alias.

URP_EXCEPTION
The analyzer found that the request did not conform to the default format. A reason code is supplied as follows:

1 The length of the resource was less than 6. (The shortest possible resource specification is /A/B/C, asking for program C to be run under transaction B with converter A.) This response and reason are the ones used when the incoming request is not an HTTP request.

2 The resource specification did not begin with a "/".

3 The resource specification contained one "/", but fewer than three of them.

4 The length of the converter name in the resource specification was 0 or more than 8.

5 The length of the transaction name in the resource specification was 0 or more than 4.

6 The length of the CICS program name in the resource specification was 0 or more than 8.

7 A "?" was found, but the length of the user token following it was 0 or more than 8.

URP_INVALID
The eye-catcher was invalid. This is an internal error.

Reference information for the converter

This section provides:
- Reference information for the **Decode** function of the converter
- Reference information for the **Encode** function of the converter

The names of the parameters and constants in the communication area passed to the converter, translated into appropriate forms for the different programming languages supported, are defined in files supplied as part of the CICS Web Interface. The files for the various languages are listed in the following table.

Language	Parameters file	Constants file
Assembler	DFHWBCDD	DFHWBUCD
C	DFHWBCDH	DFHWBUCH
COBOL	DFHWBCDO	DFHWBUCO
PL/I	DFHWBCDL	DFHWBUCL

These files give language-specific information about the data types of the fields in the communication area. If you use these files you must specify XOPTS(NOLINKAGE) on the Translator step; failure to do this causes the compile

to fail.

Decode

Summary of parameters

In the following table, the names of the parameters are given in abbreviated form: each name in the table must be prefixed with **decode_** to give the name of the parameter.

Table 3. Parameters for **Decode**

Input decode_	Inout decode_	Output decode_
client_address **client_address_string** **eyecatcher** **function** **http_version_length** **http_version_ptr** **method_length** **method_ptr** **request_header_length** **request_header_ptr** **resource_length** **resource_ptr** **user_data_length** **user_data_ptr**	**data_ptr** **server_program** **user_token**	**input_data_len** **output_data_len** **reason** **response**

Function

If the analyzer specified a converter name for the request, **Decode** is called by the alias before it calls the CICS program to service the request. It must set up the communication area for the program.

Parameters

decode_client_address
(Input only)

The 32-bit IP address of the client.

decode_client_address_string
(Input only)

The IP address of the client in dotted decimal format.

decode_data_ptr
(Input and output)

On input, a pointer to the request from the client, as modified by the analyzer.

On output, a pointer to the communication area to be passed to the CICS program.If this pointer is altered to address another storage location, the **Decode** program is responsible for FREEMAINing the original storage.

decode_eyecatcher
(Input only)

A string of length 8. Its value for **Decode** is “>decode”.

decode_function
(Input only)

A code indicating that **Decode** is being called. The value is URP_DECODE.

decode_http_version_length
(Input only)

The length in bytes of the string identifying the HTTP version supported by the client. If the request is not an HTTP request, this length is zero.

decode_http_version_ptr
(Input only)

A pointer to the string identifying the HTTP version supported by the client. If the analyzer modified this part of the request, the changes are visible here. If **decode_http_version_length** is zero, do not use this pointer.

decode_input_data_len
(Input and output)

The value to be used for the DATALENGTH parameter of the internally executed EXEC CICS LINK command when linking to the CICS program. The default value, if the value is not reset on output, is equal to the total length of the incoming request. This value should be reset to reflect any changes made to the commarea by the converter.

decode_method_length
(Input only)

The length in bytes of the method specified in the HTTP request. If the request is not an HTTP request, this length is zero.

decode_method_ptr
(Input only)

A pointer to the method specified in the HTTP request. If the analyzer modified this part of the request, the changes are visible here. If **decode_method_length** is zero, do not use this pointer.

decode_output_data_len
(Output only)

The value to be used for the LENGTH option of the EXEC CICS LINK command for the CICS program. The default value if this output is not set is 32K.

decode_reason
(Output only)

A reason code—see “Responses and reason codes” on page 36.

decode_request_header_length
(Input only)

The length of the first HTTP header in the HTTP request. If the request is not an HTTP request, this length is zero.

decode_request_header_ptr
(Input only)

A pointer to the first HTTP header in the HTTP request. If the analyzer modified this part of the request, the changes are visible here. If **decode_request_header_length** is zero, do not use this pointer.

decode_resource_length
(Input only)

The length in bytes of the string identifying the HTTP absolute path specified in the HTTP request. If the request is not an HTTP request, this length is zero.

decode_resource_ptr
(Input only)

A pointer to the string identifying the HTTP absolute path specified in the HTTP request. If the analyzer modified this part of the request, the changes are visible here. If **decode_resource_length** is zero, do not use this pointer.

decode_response
(Output only)

A response—see "Responses and reason codes".

decode_server_program
(Input and output)

A string of length 8. On input, the value supplied by the analyzer in **wbra_server_program**. On output, the name of the CICS program that is to service the request. The CICS program name must be set here or in the analyzer, or no CICS program will be called.

decode_user_data_length
(Input only)

The length in bytes of the user data for this HTTP request. If the analyzer modified this value it is visible here. If there is no user data in the request, the length is zero. If the request is not an HTTP request, this length is the length of the request.

decode_user_data_ptr
(Input only)

A pointer to any user data for this HTTP request. If the analyzer modified this part of the request, the changes are visible here. If there is no user data in the request, the pointer is zero. If the request is not an HTTP request, this pointer has the same value as **decode_data_ptr**.

decode_user_token
(Input and output)

A 64-bit token. On input, the user token supplied by the analyzer as **wbra_user_token**.On output, a token that is passed to **Encode** as **encode_user_token**.

Responses and reason codes

You must return one of the following values in **decode_response**:

URP_OK
The alias links to the CICS program using the communication area provided by **Decode**.

URP_EXCEPTION
The CICS program is not executed. The alias's action depends on the reason code:

- URP_SECURITY_FAILURE—the alias writes an exception trace entry (trace point A119), and issues a message (DFHWB0121). If the request is an HTTP request, response 403 is sent to the web browser. If the request is not an HTTP request, no response is sent, and the TCP/IP for MVS socket is closed.

- URP_CORRUPT_CLIENT_DATA—the alias writes an exception trace entry (trace point A118), and issues a message (DFHWB0121). If the request is an HTTP request, response 400 is sent to the web browser. If the request is not an HTTP request, no response is sent, and the TCP/IP for MVS socket is closed.
- Any other value—the alias writes an exception trace entry (trace point A11A), and issues a message (DFHWB0121). If the request is an HTTP request, response 501 is sent to the web browser. If the request is not an HTTP request, no response is sent, and the TCP/IP for MVS socket is closed.

URP_INVALID
The CICS program is not executed. The alias writes an exception trace entry (trace point A11B), and issues a message (DFHWB0121). If the request is an HTTP request, response 501 is sent to the web browser. If the request is not an HTTP request, no response is sent, and the TCP/IP for MVS socket is closed.

URP_DISASTER
The CICS program is not executed. The alias writes an exception trace entry (trace point A11C), and issues a message (DFHWB0121). If the request is an HTTP request, response 501 is sent to the web browser. If the request is not an HTTP request, no response is sent, and the TCP/IP for MVS socket is closed.

If you return any other value in **decode_response**, the CICS program is not executed. The alias writes an exception trace entry (trace point A11D), and issues a message (DFHWB0121). If the request is an HTTP request, response 500 is sent to the web browser. If the request is not an HTTP request, no response is sent, and the TCP/IP for MVS socket is closed.

You may supply a 32-bit reason code in **decode_reason** to provide further information in error cases. Except as noted above, the CICS Web Interface does not take any action on the reason code returned by **Decode**, except as indicated above under URP_EXCEPTION. The reason code is output in any trace entry that results from the invocation of **Decode**.

Encode

Summary of parameters

In the following table, the names of the parameters are given in abbreviated form: each name in the table must be prefixed with **encode_** to give the name of the parameter.

Table 4. Parameters for **Encode**

Input encode_	Inout encode_	Output encode_
eyecatcher **function** **input_data_len** **user_token**	**data_ptr**	**reason** **response**

Function

If the analyzer specified a converter name for the request, **Encode** is called by the alias after the CICS program has ended. It constructs the response from the contents of the communication area.

Parameters

encode_data_ptr
(Input and output)

On input, a pointer to the communication area returned by the CICS program. If no CICS program was called, it is a pointer to the communication area created by **Decode**. On output, a pointer to the buffer containing the response to be sent to the client. The buffer must be doubleword aligned. The first four bytes must be a 32-bit unsigned number specifying the length of the buffer. (In COBOL, specify this as PIC 9(8) COMP.) The rest of the buffer is the response. If this pointer is altered to address another storage location, the **Encode** program is responsible for FREEMAINing the original storage.

encode_eyecatcher
(Input only)

A string of length 8. Its value for **Encode** is ">encode".

encode_function
(Input only)

A code indicating that **Encode** is being called. The value is URP_ENCODE.

encode_input_data_len
(Input only)

The length of the communication area as specified by **Decode** in **decode_output_data_len**.

encode_reason
(Output only)

A reason code—see "Responses and reason codes" on page 39.

encode_response
(Output only)

A response—see "Responses and reason codes" on page 39.

encode_user_token
(Input only)

The 64-bit token output by **Decode** as **decode_user_token**.

Responses and reason codes

You must return one of the following values in **encode_response**:

URP_OK
The response in the buffer pointed to by **encode_data_ptr** is sent to the client.

URP_EXCEPTION
The alias's action depends on the reason code:

- URP_SECURITY_FAILURE—the alias writes an exception trace entry (trace point A119), and issues a message (DFHWB0122). If the request is an HTTP request, response 403 is sent to the web browser. If the request is not an HTTP request, no response is sent, and the TCP/IP for MVS socket is closed.
- URP_CORRUPT_CLIENT_DATA—the alias writes an exception trace entry (trace point A118), and issues a message (DFHWB0122). If the request is an HTTP request, response 400 is sent to the web browser. If the request is not an HTTP request, no response is sent, and the TCP/IP for MVS socket is closed.
- Any other value—the alias writes an exception trace entry (trace point A11A), and issues a message (DFHWB0122). If the request is an HTTP request, response 501 is sent to the web browser. If the request is not an HTTP request, no response is sent, and the TCP/IP for MVS socket is closed.

URP_INVALID
The alias writes an exception trace entry (trace point A11B), and issues a message (DFHWB0122). If the request is an HTTP request, response 501 is sent to the web browser. If the request is not an HTTP request, no response is sent, and the TCP/IP for MVS socket is closed.

URP_DISASTER
The alias writes an exception trace entry (trace point A11C), and issues a message (DFHWB0122). If the request is an HTTP request, response 501 is sent to the web browser. If the request is not an HTTP request, no response is sent, and the TCP/IP for MVS socket is closed.

If you return any other value in **encode_response**, the alias writes an exception trace entry (trace point A11D), and issues a message (DFHWB0122). If the request is an HTTP request, response 501 is sent to the web browser. If the request is not an HTTP request, no response is sent, and the TCP/IP for MVS socket is closed.

You may supply a 32-bit reason code in **encode_reason** to provide further information in error cases. Except as noted above, the CICS Web Interface does not take any action on the reason code returned by **Encode**. The reason code is output in any trace entry that results from the invocation of **Encode**.

Chapter 4. Writing CICS applications for the CICS Web Interface

Many existing CICS programs can be used without change to process requests, provided that they are supported by a converter. The programs must have been written to be independent of a principal facility. The converter must translate the incoming request into the communication area that the CICS program is expecting, and must translate the output communication area into an appropriate response.

This chapter is concerned with facilities provided by the CICS Web Interface to help you write new CICS application programs that exploit the kind of environment information present in HTTP requests and the power of HTML for presenting information. The chapter contains the following sections:

- "Writing CICS programs to process HTTP requests" gives guidance about the design of new CICS programs to handle HTTP requests.
- "Reference information for the HTML template manager" on page 42 describes the HTML template manager, and how to use it.
- "Reference information for the environment variables program" on page 49 describes the environment variables program, and how to use it.
- "Reference information for the state management sample programs" on page 51 describes the state management program, and how to use it.
- "Reference information for the parser program" on page 53 describes the parser for HTML forms data, and how to use it.

This chapter contains Product-sensitive Programming Information and Associated Guidance Information.

Writing CICS programs to process HTTP requests

The CICS program communicates with the CICS Web Interface by means of a CICS communication area. If the program is supported by a converter, the communication area contains the information put in it by **Decode**, otherwise it contains the entire HTTP request. The HTTP header information is in EBCDIC, and if the analyzer asks for data conversion, the user data has been translated using the analyzer-specified key.

You may use the environment variable program DFHWBENV to retrieve the following information present in the HTTP request:

- The IP address of the client
- The IP address of the host
- The local host name
- The HTTP method
- The HTTP version

You may also retrieve an indicator of the CICS release under which the program is running. See "Reference information for the environment variables program" on page 49 for more information about DFHWBENV and the format in which it presents its output.

You can use this information and the information in the communication area to control processing in your CICS program. You should restrict yourself to the DPL subset of the CICS application programming interface. If you wish to write a

program that can be executed with or without a principal facility, you should make it able to handle errors that arise from the use of commands outside the DPL subset. The DPL subset is documented in *CICS for MVS/ESA Application Programming Reference*.

APAR PQ08889

Responses can be greater than 32k now.

If you wish to use HTML in your response, you can use the HTML template manager DFHWBTL to insert templates in the HTTP response, and to replace symbols in the templates with values that you specify. See "Reference information for the HTML template manager" for more information about the HTML template manager and its operation.

The storage containing the response must begin with a 32-bit integer specifying the length of the response plus 4 for the integer. You can build the HTTP response in the communication area, in which case the maximum length of the response is 4 less than the length of the communication area.

If your program is operating under the CICS Web Interface or under the CICS business logic interface in pointer mode, you can build the response in any area of storage other than the communication area, provided that you pass the address of the storage to Encode in the communication area. In this way you can build HTTP responses longer than 32K.

If your program is operating under the CICS business logic interface in offset mode, you can build the response only in the communication area provided.

The response can be constructed entirely by the CICS program, or partly by the CICS program and partly by Encode. Translation of the various parts of the response from EBCDIC to ASCII (for the headers) and to the client code page (for the user data) is dealt with by the alias program.

A sample application program

DFH$WB1A is a sample program provided with the CICS Web Interface. It uses no converter, and constructs a simple HTTP response whose body is an HTML page. The sample program can be run by enabling the CICS Web Interface with the default analyzer DFHWBADX, and by entering a suitable URL with the absolute path `/CICS/CWBA/DFH$WB1A` with a web browser. The response displays the message "`DFH$WB1A on system XXXXXXXX successfully invoked through the CICS Web Interface.`" with `XXXXXXXX` replaced by the application ID of the CICS system in which the CICS Web Interface is running.

Reference information for the HTML template manager

The HTML template manager helps you to write CICS application programs that create HTML pages to be sent to an HTTP client. You use EXEC CICS LINK to call DFHWBTL. An HTML page can be built from one or more templates. The templates can be read from an MVS partitioned data set (PDS), or can be provided inline in your application program. Templates can contain HTML symbols, and the template manager replaces the symbols with values from a symbol table as it adds

the template to a page. The template manager allows you to set up and modify a symbol table as you add templates to the HTML page.

The functions of the template manager are summarized as follows:

- BUILD_HTML_PAGE combines the functions of START_HTML_PAGE, ADD_HTML_TEMPLATE, and END_HTML_PAGE.
- START_HTML_PAGE establishes an environment for the next three functions, and allows you to put values in the symbol table.
- ADD_HTML_SYMBOLS adds symbols to the symbol table. It also modifies the values of symbols already defined in the symbol table.
- ADD_HTML_TEMPLATE adds a template to the HTML page, replacing symbols in the template with the values defined in the symbol table.
- END_HTML_PAGE destroys the environment established in START_HTML_PAGE.

You call the template manager using EXEC CICS LINK as follows:

```
EXEC CICS LINK PROGRAM(DFHWBTL) COMMAREA(...) LENGTH(...)
```

You supply the communication area addressed by the COMMAREA option of the command. The contents of the communication area are described below.

In this chapter the various program elements (values) are given symbolic names. These names, translated into appropriate forms for the different programming languages supported, are defined in files supplied as part of the CICS Web Interface. The files for the various languages are as follows:

Language	**File**
Assembler	DFHWBTLD
C	DFHWBTLH
COBOL	DFHWBTLO
PL/I	DFHWBTLL

These files give language-specific information about the data types of the fields in the communication area.

Parameters in the communication area

APAR PQ07604

Added wbtl_template_abstime and changed order of html_buffer and symbol_list parameters, RGW 5 Nov 1997

The following table summarizes the use of the parameters by function.

Table 5. Parameters for the HTML template manager

Function	Parameters
WBTL_START_HTML_PAGE	**wbtl_version_no** input **wbtl_function** input **wbtl_response** output **wbtl_reason** output **wbtl_connect_token** inout **wbtl_symbol_list_ptr** input **wbtl_symbol_list_len** input
WBTL_ADD_HTML_SYMBOLS	**wbtl_version_no** input **wbtl_function** input **wbtl_response** output **wbtl_reason** output **wbtl_connect_token** input **wbtl_symbol_list_ptr** input **wbtl_symbol_list_len** input
WBTL_ADD_HTML_TEMPLATE	**wbtl_version_no** input **wbtl_function** input **wbtl_response** output **wbtl_reason** output **wbtl_connect_token** input **wbtl_template_name** input **wbtl_template_abstime** output **wbtl_template_buffer_ptr** input **wbtl_template_buffer_len** input **wbtl_html_buffer_ptr** inout **wbtl_html_buffer_len** inout
WBTL_END_HTML_PAGE	**wbtl_version_no** input **wbtl_function** input **wbtl_response** output **wbtl_reason** output **wbtl_connect_token** input
WBTL_BUILD_HTML_PAGE	**wbtl_version_no** input **wbtl_function** input **wbtl_response** output **wbtl_reason** output **wbtl_template_name** input **wbtl_template_abstime** output **wbtl_template_buffer_ptr** input **wbtl_template_buffer_len** input **wbtl_symbol_list_ptr** input **wbtl_symbol_list_len** input **wbtl_html_buffer_ptr** inout **wbtl_html_buffer_len** inout

wbtl_version_no
(Input only)

The version number of the template manager interface. Specify zero.

wbtl_function
(Input only)

Specify the function you wish to perform as one of the following:
- WBTL_BUILD_HTML_PAGE
- WBTL_START_HTML_PAGE

- WBTL_ADD_HTML_SYMBOLS
- WBTL_ADD_HTML_TEMPLATE
- WBTL_END_HTML_PAGE

wbtl_response
(Output only)

The response from the template manager to the function and inputs. See "Responses and reason codes" on page 46.

wbtl_reason
(Output only)

Might contain additional information about an error for some responses. See "Responses and reason codes" on page 46.

wbtl_connect_token
(Input and output)

As input to WBTL_START_HTML_PAGE, this token must be zero.

As output from WBTL_START_HTML_PAGE, this token represents the page environment established by WBTL_START_HTML_PAGE, and you must save it for use with other functions. You can have several tokens in use at once, and the template manager maintains separate page environments for each token.

As input to WBTL_ADD_HTML_SYMBOLS, WBTL_ADD_HTML_TEMPLATE, and WBTL_END_HTML_PAGE, this the token identifies the HTML page environment.

wbtl_template_name
(Input only)

As optional input to WBTL_BUILD_HTML_PAGE, and WBTL_ADD_HTML_TEMPLATE, this is an eight-character field, padded on the right with spaces. If you want the template manager to use a template from the PDS, put the name of the member here. If you want the template manager to use an inline template, put spaces here and use the **wbtl_template_buffer_ptr** and **wbtl_template_buffer_len** fields.

APAR PQ07604
Added wbtl_template_abstime and changed order of html_buffer and symbol_list parameters, RGW 5 Nov 1997

wbtl_template_abstime
(output only)

Output from WBTL_ADD_HTML_TEMPLATE and WBTL_BUILD_HTML_PAGE when the template manager is requested to use the PDS member specified by wbtl_template_name. This is the date and time (in CICS ABSTIME format) when the template was last modified, if the modification was made with the ISPF editor. Otherwise it is the current date and time.

wbtl_template_buffer_ptr
(Input only)

As optional input to WBTL_BUILD_HTML_PAGE and WBTL_ADD_HTML_TEMPLATE, this is the address of the template to be used. If you want the template manager to use an inline template, use this

field. If you want the template manager to use a template from the PDS, do not use this field, but use **wbtl_template_name** instead.

wbtl_template_buffer_len
(Input only)

As optional input to WBTL_BUILD_HTML_PAGE and WBTL_ADD_HTML_TEMPLATE, this is the length in bytes of the template pointed to by **wbtl_template_buffer_ptr**. If you want the template manager to use an inline template, use this field. If you want the template manager to use a template from the PDS, do not use this field, but use **wbtl_template_name** instead.

wbtl_symbol_list_ptr
(Input only)

This field is a required input to WBTL_ADD_HTML_SYMBOLS, and an optional input to WBTL_BUILD_HTML_PAGE and WBTL_START_HTML_PAGE. It is the address of the list of symbols to be used to update the symbol table. The format of the list is described in "Symbols, symbol table, and symbol lists" on page 47. If the function is WBTL_ADD_HTML_SYMBOLS, you must use the **wbtl_connect_token** to identify the page environment whose symbol table is to be updated.

wbtl_symbol_list_len
(Input only)

This field is a required input to WBTL_ADD_HTML_SYMBOLS, and an optional input to WBTL_BUILD_HTML_PAGE and WBTL_START_HTML_PAGE. It is the length in bytes of the list of symbols to be used to update the symbol table.

wbtl_html_buffer_ptr
(Input and output)

As input to WBTL_BUILD_HTML_PAGE and WBTL_ADD_HTML_TEMPLATE, this field is the address of the unused portion of the buffer that contains the HTML page being constructed. As output from WBTL_BUILD_HTML_PAGE and WBTL_ADD_TEMPLATE, this field is the address of the remaining space in the buffer.

wbtl_html_buffer_len
(Input and output)

As input to WBTL_BUILD_HTML_PAGE and WBTL_ADD_HTML_TEMPLATE, this is the length in bytes of the unused portion of the buffer that contains the HTML page being constructed. As output from WBTL_BUILD_HTML_PAGE and WBTL_ADD_HTML_TEMPLATE, this is the length in bytes of the remaining space in the buffer.

Responses and reason codes

WBTL_OK
The operation ended successfully.

WBTL_EXCEPTION
The template manager detected an error in the operation. The following reason values are possible:

WBTL_TEMPLATE_NOT_FOUND
The template manager could not find the template named in **wbtl_template_name** in the PDS.

WBTL_TEMPLATE_TRUNCATED
There was not enough room left in the buffer for the template. The HTML template manager has used all the space available, and discarded the rest of the template.

WBTL_INVALID
The template manager detected an error in the parameters in the communication area. The following reason values are possible:

WBTL_INVALID_BUFFER_PTR
The value in **wbtl_html_buffer_ptr** was zero when an address was required.

WBTL_INVALID_FUNCTION
The value in **wbtl_function** was not recognized.

WBTL_INVALID_SYMBOL_LIST
An input symbol list was required, but either **wbtl_symbol_list_ptr** was zero, or **wbtl_symbol_list_len** was zero.

WBTL_INVALID_TOKEN
The operation was expecting an input **wbtl_connect_token**, but found its value was zero. All tokens output by the HTML template manager are non-zero.

WBTL_DISASTER
The template manager detected an unrecoverable error. The following reason values are possible:

WBTL_FREEMAIN_ERROR
There was an error while attempting to release storage.

WBTL_GETMAIN_ERROR
There was an error while attempting to acquire storage.

Symbols, symbol table, and symbol lists

This section describes the symbols in an HTML template, and how the HTML template manager uses the symbol table to replace the symbols with values.

Symbols in an HTML template

In an HTML template, symbols begin with an ampersand (“&”) and end with a semicolon (“;”), and contain up to 32 characters with no imbedded spaces. Thus the following template contains `&mytitle;` as its only symbol.

```
<html>
  <head>
    <title>
      &mytitle;
    </title>
  </head>
  <body>
```

Symbol lists

The template manager maintains a symbol table for each active page environment. In WBTL_BUILD_HTML_PAGE and WBTL_ADD_HTML_TEMPLATE, the template manager uses the input symbol list, if any, to create or update the symbol table, and then replaces the symbols in the template by their values in the table.

A symbol list is a character string consisting of one or more definitions with single byte separators. By default, the single byte separator is an ampersand, but the caller of the template manager may choose their own separator. The method for doing this is described later in this section. A definition consists of a name, an equals sign, and a value. Here is an example:

```
mytitle=New Authors&auth1=Halliwell Sutcliffe&auth2=Stanley Weyman
```

The *name* must contain only uppercase and lowercase letters, numbers, and underscores ("_"). The name is case-sensitive, so uppercase letters are regarded as different from lowercase letters. Unlike the symbols in the template, the names in the symbol list have neither an ampersand at the beginning, nor a semicolon at the end. The symbol `&mytitle;` in the template corresponds to the name `mytitle` in the symbol list.

You may specify your own symbol separator to override the default of ampersand. To do this, insert the character sequence '&DELIM=x&' at the start of the symbol list. The x is a single byte separator which is used in the list of symbols which follows. This single byte may be any hexadecimal value apart from a space (X'40') or an equals sign (X'7E'). If a space or an equals sign is specified, the template manager disregards it and assumes that an ampersand is being used as the separator. If a valid separator override is specified, the application must use it to separate symbol values from the symbol names which follow them. The application must also guarantee that the separator does not appear anywhere in a symbol value. It must also ensure that WBTL_SYMBOL_LIST_LEN includes the length of '&DELIM=x&'.

If an application specifies a valid separator override, all symbol values in the symbol list for this call to the template manager are treated as unescaped character sequences. This means that they are substituted into the template without undergoing any conversion. For example, a plus sign ('+') remains as a plus, and sequences such as %2B remain as they are rather than being converted to single characters.

If the application does not specify its own valid separator, the following rules apply to symbol values:

- The *value* in the symbol list can have any characters except ampersand, but with some restrictions on the use of the percent sign ("%") and the plus sign ("+"). A percent sign must be followed by two characters that are hexadecimal digits. When the value is put into the symbol table, a plus sign is interpreted as a space, a percent sign and the two hexadecimal digits following it are interpreted as the EBCDIC equivalent of the single ASCII character denoted by the two digits, and the remaining characters are left as they are. If you want a plus sign in the value in the symbol table, you must put `%2B` in the value in the symbol list. If you want a percent sign in the value in the symbol table, you must put in the value `%25` in the symbol list. If you want an ampersand in the value in the symbol table, you must put `%26` in the value in the symbol list. If you want a space in the value in the symbol table, the value in your symbol list may contain a space, a plus sign, or `%20`.

Operational example

The following symbol list

```
mytitle=New Authors&auth1=Halliwell Sutcliffe&auth2=Stanley Weyman
```

provides definitions of three symbols. Note that an ampersand is a separator that separates a name from the following value, and is not part of the name that

follows it. In an HTML template, &mytitle; is replaced by New Authors, &auth1; by Halliwell Sutcliffe, and &auth2; by Stanley Weyman.

The following is an example of how an application specifies its own separator:

```
&DELIM=!&COMPANY=BLOGGS & SON!ORDER=NUTS AND BOLTS
```

In this example, the symbol COMPANY has a value of 'BLOGGS & SON', and the symbol ORDER has a value of 'NUTS AND BOLTS'. The delimiter is an exclamation mark, but it is better to use a non-printable character which will not appear in the rest of the symbol string.

Using the output of the environment variables program

The output of the environment variables program, described in "Reference information for the environment variables program", can be used as a symbol list for the HTML template manager. If you want to use an environment variable that is derived from one of the HTTP headers, you cannot always predict whether it will appear in the environment variables string. Therefore, you should always initialize the symbol table so that names that represent environment variables are associated with default values. Then you can use the output from the environment variables program as a symbol list. For example, if you want to use the &HTTP_REFERER; and &HTTP_AUTHORIZATION; variables in your template, but you do not know whether the client has set them, you could pass the following symbol string to the template manager first:

```
HTTP_REFERER=&HTTP_AUTHORIZATION=
```

This associates both the names with a null string value in the symbol table.

Reference information for the environment variables program

The environment variables program is DFHWBENV. It extracts information about the server (the CICS region in which the server controller is running), and the client (the web browser that sent the current request). You can use EXEC CICS LINK to call it. You must supply a communication area that is long enough to contain the expected response. The exact length of the response depends on the nature of your connection with the client, and the values set by the client's browser program, but 1024 bytes will usually be enough. On return, the communication area contains a 32-bit integer followed by a sequence of values of environment variables. The 32-bit integer specifies the length of the string that follows it. The values are specified with the following format:

```
variable-name=value
```

Each value is separated from the following variable name by an ampersand. None of the values contain an ampersand. This format is the same as that required for input as a symbol list to the HTML template manager (DFHWBTL), and to the parser (DFHWBPA). If the environment variables program cannot return any variables, it returns a length of zero. If the communication area you provide is not long enough to contain all the variables and their values, the program abends with abend code AWBC.

Note that DFHWBENV can be linked to only from the alias transaction; it cannot be linked to from the analyzer.

The meaning of the value for each variable name that can occur in the communication area is as follows:

CONTENT_LANGUAGE
The national language of any user data in the HTTP request. The value contains the ISO 3316 language code, optionally qualified by an ISO 639 country code. It is extracted from the Content-Language HTTP header. If there is no Content-Language header, the value is a null string.

CONTENT_LENGTH
The character representation of the decimal length of any user data in the HTTP request. It is extracted from the Content-Length HTTP header. If there is no user data, the value is zero.

CONTENT_TYPE
The MIME format of any user data in the HTTP request. It is extracted from the Content-Type HTTP header. If there is no user data, the value is a null string.

HTTP_ACCEPT
The contents of all the Accept HTTP headers, separated by commas. These values represent the MIME types that the browser is prepared to accept, so the list should never be empty. However, if there are no Accept headers, the value is a null string.

HTTP_ACCEPT_ENCODING
The contents of the Accept-Encoding HTTP header. If there is no Accept-Encoding header, the variable is not returned.

HTTP_ACCEPT_LANGUAGE
The contents of the Accept-Language HTTP header. If there is no Accept-Language header, the variable is not returned.

HTTP_AUTHORIZATION
The contents of the Authorization HTTP header. If there is no Authorization header, the variable is not returned.

HTTP_CHARGE_TO
The contents of the Charge-To HTTP header. If there is no Charge-To header, the variable is not returned.

HTTP_FROM
The contents of the From HTTP header. If there is no From header, the variable is not returned.

HTTP_IF_MODIFIED_SINCE
The contents of the If-Modified-Since HTTP header. If there is no If-Modified-Since header, the variable is not returned.

HTTP_PRAGMA
The contents of the Pragma HTTP header. If there is no Pragma header, the variable is not returned.

HTTP_REFERER
The contents of the Referer HTTP header. If there is no Referer header, the variable is not returned.

HTTP_USER_AGENT
The contents of the User-Agent HTTP header. This is the product name of the web browser program. If there is no User-Agent header, the variable is not returned.

QUERY_STRING
The query string from the HTTP request. Any ampersands in the query

string are expanded to %26, and any equals signs are expanded to %3D. If there is no query string, the value is a null string.

REMOTE_ADDR
The IP address of the client in dotted decimal format.

REMOTE_HOST
The fully-qualified name of the client, if this can be obtained from the name server. If the name cannot be found, or 'Enable DNS' has been set to NO in the connection manager (see "Chapter 5. Configuration with the connection manager" on page 55), the value is a null string.

REMOTE_USER
The user ID that has been assigned to the current request.

REQUEST_METHOD
The method name specified in the first HTTP header received from the client. It is one of "GET", "POST", "HEAD", "SHOWMETHOD", "PUT", "DELETE", "LINK", "UNLINK".

APAR PQ11796
Name Server information updated

SERVER_NAME
The fully-qualified name of the connection, for example "www.hursley.ibm.com". If CICS was unable to obtain its own name from the domain name server when the CICS Web Interface was enabled, the dotted decimal address of the connection will be returned instead.

SERVER_PORT
The character representation of the decimal value of the TCP/IP port on which the request was received, for example "80".

SERVER_PROTOCOL
The name of the Internet protocol describing the data received, usually "HTTP/1.0".

SERVER_SOFTWARE
The name and version of the CICS product.

Reference information for the state management sample programs

Two state management sample programs, DFH$WBST and DFH$WBSR are supplied with the CICS Web Interface. They allow a transaction to save data for later retrieval by the same transaction, or by another transaction. The saved data is accessed by a token that is created by the state management program for the first transaction. The first transaction must pass the token to the transaction that is to retrieve the data. DFH$WBST uses EXEC CICS GETMAIN to allocate storage for the saved data. DFH$WBSR saves the data in temporary storage queues, one for each token, so that, with suitable temporary storage table definitions, the data can be accessed from several CICS systems. The rest of this section applies equally to either program.

The state management program and the tokens it allocates can be used in many ways. Here are two suggestions:

- The token can be used as a *conversation token*, that is a token that identifies information that is to preserved throughout a pseudoconversation. A

conversation token can be managed by the converter or the CICS program, and is best conveyed from program to program in a pseudoconversation as a hidden field in an HTML form.

- The token can be used as a *session token*, that is a token that identifies information that is to preserved throughout an extended interaction between an end-user and various CICS programs, perhaps over several pseudoconversations. A session token can be managed by the analyzer, and is best conveyed from interaction to interaction as a query string in a URL. This use of a state management token is illustrated by the security analyzer, security converter, and security sign-on sample programs described in "The security sample programs" on page 66.

The state management program provides the following operations:

- Create a new token.
- Store information and associate it with a previously-created token.
- Retrieve information previously associated with a token.
- Destroy information associated with a token, and invalidate the token.
- Remove information and tokens that have expired.

The last operation is an internal operation, not explicitly invoked by the caller.

If DFH$WBST or DFH$WBSR is invoked from transaction CWBP, all state data is deleted, and any storage associated with it is FREEMAINed. There is no CICS-supplied definition for transaction CWBP, so if you want to use it you must first define it yourself, specifying DFH$WBST or DFH$WBSR as its associated program

The layout of the 268-byte communication area is shown in the following table. You must clear the communication area to binary zeroes before setting the inputs for the function you require.

Table 6. Parameters for the state management program

Offset	Length	Type	Value	Notes
0	4	C		Eyecatcher
4	1	C	'C' 'R' 'S' 'D'	Create Retrieve Store Destroy This is the function code. It is a required input to every call.
5	1	X		Return code. This is an output from every call.
6	2	X		Reserved
8	4	F		Token. This is an output from a Create call, and an input to every other call.
12	256	C		User data. This is an input to a Create call, and an output from a Retrieve call. It is not used in other calls.

The return codes are as follows:

0 The requested function was performed.

- If the function was Create, a new token is available at offset 8.

- If the function was Retrieve, the user data associated with the input token at offset 8 is now in the user data area at offset 12.
- If the function was Store, the input user data at offset 12 is now associated with the input token and offset 8. Any user data previously associated with the token is overwritten.
- If the function was Destroy, the data associated with the input token at offset 8 has been discarded, and the token is no longer valid.

2 The function code at offset 4 was not valid. Correct the program that sets up the communication area.

3 The function was Create, but EXEC CICS GETMAIN gave an error response.

4 The function was Retrieve, Store, or Destroy, but the input token at offset 8 was not found. Either the input token is not a token returned by Create, or it has expired.

5 EXEC CICS WRITEQ TS gave an error response when writing internal data to a temporary storage queue.

7 EXEC CICS ASKTIME gave an error response.

8 EXEC CICS READQ TS gave an error response when reading internal data from a temporary storage queue.

9 EXEC CICS ASKTIME gave an error response during timeout processing.

11 The function was Create, but EXEC CICS WRITEQ TS gave an error response. This return code is produced only by DFH$WBSR.

12 The function was Retrieve, but EXEC CICS READQ TS gave an error response. This return code is produced only by DFH$WBSR.

13 The function was Store, but EXEC CICS WRITEQ TS gave an error response. This return code is produced only by DFH$WBSR.

14 The function was Destroy, but EXEC CICS DELETEQ TS gave an error response. This return code is produced only by DFH$WBSR.

Reference information for the parser program

The CICS Web Interface parser program is DFHWBPA. It parses strings of the form:

key1=value1&key2=value2&key3=value3 ...

key1 is a keyword, *value1* is the corresponding value, and so on. The keyword/value pairs must be separated by ampersands as shown in the example. If there is only one keyword/value pair there must be no ampersand.

- A *keyword* must contain only upper and lower case letters, digits, and underscores ("_"). It must not contain any imbedded blanks.
- A *value* can contain any character except an ampersand.

The kinds of strings that the parser accepts are the same as:
- Data transmitted by HTTP clients as query strings
- Forms data from HTTP clients
- Output from the environment variables program DFHWBENV
- Input to the HTML template manager.

The parser accepts a string and a keyword as input, and returns the corresponding value as output. If the string does not contain the keyword, the output is nulls.

APAR 16019 and 16688
Removed the restriction: “The communication area must not be more that 4096 bytes long.”

The program is called by EXEC CICS LINK. You supply a communication area containing the keyword to be found, two ampersands, and the string to be searched.

```
EXEC CICS LINK PROGRAM(DFHWBPA) COMMAREA(...) LENGTH(...)
```

The commarea length should be equal to the length of the string plus the keyword prefix. If the length is greater than this, an ampersand should be used to mark the logical end of string.

When the parser returns to your program, the communication area contains the value followed by nulls.Any escaped characters will be unescaped.

The following example illustrates the operation of the parser. Suppose the input communication area contains the following string:

```
a1&&myt=New Authors&a1=Halliwell Sutcliffe&a2=Stanley Weyman
```

The output is

```
Halliwell Sutcliffe
```

padded to 60 bytes (the length of the input communication area) with nulls.

Unescaping considerations

The analyzer determines if the parser is to unescape or not. The analyzer specifies this in the **wbra_unescape** parameter.

It is recommended that the decision on whether to perform unescaping is made in the analyzer program, and not deferred until execution of the converter or target program.

If **wbra_unescape** has been set to **wbra_unescape_not_required**, the parser assumes that the data passed to it is still escaped, and unescapes the value it returns. Plus signs are converted to spaces.

If **wbra_unescape** has been set to **wbra_unescape_required** or **wbra_unescape_completed**, the parser assumes that the data passed to it has already been unescaped and does not attempt to unescape it again. Ampersands are not handled correctly.

If the data passed to the parser is in the wrong format, the following problems can occur:

- If the data is unescaped and the parser expects it to be escaped, ampersands, plus signs, and percent signs followed by two hexadecimal digits are not handled correctly.
- If the data is escaped and the parser expects it to be unescaped, the value retruned from the parser remains unescaped.

Chapter 5. Configuration with the connection manager

This chapter is organized as follows:

- "What the connection manager is for" describes the main uses of the connection manager.
- "Starting the connection manager" on page 56 describes different ways of starting the connection manager.
- There are sections on using the connection manager panels as follows:
  - "Enabling the CICS Web Interface" on page 59 describes how to use panels to enable the CICS Web Interface.
  - "Disabling the CICS Web Interface" on page 61 describes how to use panels to disable the CICS Web Interface.
  - "Updating the CICS Web Interface data set" on page 63 describes how to use panels to update the configuration information in the CICS Web Interface data set.
  - "Updating the CICS Web Interface status" on page 64 describes how to use panels to change the current configuration of the CICS Web Interface.

What the connection manager is for

You can use the connection manager to:

- Enable the interface when it is disabled
- Modify the operating status of the interface when it is enabled
- Disable the interface when it is enabled
- Initialize the CICS Web Interface data set, or modify its contents.

When the CICS Web Interface is enabled, it can receive and process requests from TCP/IP for MVS. When it is disabled, it cannot receive or process requests. Creating or modifying the contents of the CICS Web Interface data set can be done whether the interface is enabled or disabled.

Enabling the interface

You can use the connection manager to enable the CICS Web Interface in two ways:

- Operator-assisted enable—Before you enable the interface, you can modify any or all of the operating options. The changes you make during an operator-assisted enable can be temporary, lasting only until the next time you disable the interface, or you can store them in the CICS Web Interface definition record, and use them the next time you enable the interface.
- Automatic enable—The contents of the CICS Web Interface data set determine the operating status of the interface.

Disabling the interface

There are two ways of disabling the interface: normal, and immediate. The effects of disable processing are described in "Disabling the CICS Web Interface" on page 61.

Starting the connection manager

You can start the connection manager in various ways:

- From a terminal that can display BMS maps of 80 columns and 24 rows. You can work with the connection manager panels described in this chapter.
- From a CICS console.
- Using an EXEC CICS START command.
- From a sequential terminal.
- From a PLT program.

The effect of starting the connection manager depends on:

- Whether the CICS Web Interface is enabled or disabled
- Whether you start the connection manager from a terminal that can display the connection manager's BMS maps
- Whether you enter a fast-path command (described below) with the transaction name
- Whether the Automatic Enable option in the CICS Web Interface data set is set to YES.

Fast-path commands for enabling the CICS Web Interface are described in "When the interface is disabled". Fast-path commands for disabling the CICS Web Interface are described in "When the interface is enabled" on page 57.

If you wish to write a program (including a PLT program) to use EXEC CICS START TRANSID(CWBC) to start the connection manager, and wish to specify a fast-path command, the fast path command must be addressed by the FROM option of the START command.

If you start the connection manager in a way that does not allow panels to be shown (EXEC CICS START, or a non-BMS terminal, for example) and the action is to show a panel, error message DFHWB1505 is written to the transient data destination CWBO.

When the interface is disabled

When the interface is disabled, the effect of entering the transaction name (and optional fast-path command) on a terminal that can display the connection manager's BMS maps is as follows:

CWBC
: If Automatic Enable is YES, automatic enable processing occurs, unless there is nothing in the CICS Web Interface data set. If there is nothing in the CICS Web Interface data set, or if Automatic Enable is NO, a BMS panel (DFHWB01) is shown.

CWBC E A(N)
: A BMS panel (DFHWB01) is shown. (E A(N) is a fast-path command for CWBC.)

CWBC E A(Y)
: Automatic enable processing occurs. If there is nothing in the CICS Web Interface data set, the default values of the options are stored in it. (E A(Y) is a fast-path command for CWBC.)

TCP/IP for MVS should be started before you try to enable the CICS Web Interface with the connection manager, otherwise you have to reenable the interface after starting TCP/IP for MVS.

When the interface is enabled

When the CICS Web Interface is enabled, the effect of entering the transaction name (and optional fast-path command) is as follows:

CWBC
A BMS panel (DFHWB04) is shown.

CWBC D(N)
Normal disable processing occurs. (D(N) is a fast-path command for CWBC.)

CWBC D(I)
Immediate disable processing occurs. (D(I) is a fast-path command for CWBC.)

Using the connection manager BMS panels

All leading and trailing blanks are ignored on BMS input.

At the top of all panels is a panel identifier in the right corner (for example, DFHWB02) and CWBC in the left corner.

On the bottom of all panels, the fourth line from the bottom gives the status of the CICS Web Interface, the third line from the bottom is a prompt line, while the bottom line lists the available PF keys, which can include:

PF1 Help information (all panels)
PF3 Exit CWBC (you are prompted to confirm by using PF3 again)
PF4 Write fields to the CICS Web Interface data set (only where shown)
PF7 Scroll up (only where shown)
PF8 Scroll down (only where shown)
PF9 Display messages relating to current input
PF12 Cancel this panel and return to the previous panel

Connection manager error message output

The destination of connection manager messages depends on the nature of the message, as follows:

- Severe errors requiring operator intervention are sent to the console. No other messages go to the console.
- Messages relating to invalid input on the panel can be displayed by pressing PF9.
- Messages reporting internal errors are sent to CWBO, and in most cases they can be displayed on the terminal by pressing PF9.

Using PF9 to display messages

During the operation of the connection manager, error messages might be issued. These are not displayed immediately on the screen, but a prompt appears on the prompt line to say that messages are waiting to be viewed. To see the messages, press PF9. The number and text of the messages is displayed. You can look up the messages in “Appendix B. Messages and Codes” on page 93 for more information about the error, and for advice about what to do next.

When you have read the messages, you may press Enter, PF3 or PF12 to return to the input panel.

Starting the connection manager when the CICS Web Interface is disabled

If the CICS Web Interface is disabled, panel DFHWB01 is shown. (See Figure 5.)

```
CWBC                          CICS Web Interface                          DFHWB01

Select one of the following. Then press Enter.

_  1. Enable the Web Interface
   2. View or modify the Web Interface data set









Current Status: Disabled


                                                    SYSID= ZAL1  APPLID= IYK2ZAL1
PF1=Help   PF3=Exit   PF9=Messages   PF12=Return
```

Figure 5. Panel DFHWB01

The results of selecting the various options is as follows:

1. See “Enabling the CICS Web Interface” on page 59.
2. See “Updating the CICS Web Interface data set” on page 63.

Starting the connection manager when the CICS Web Interface is enabled

If the CICS Web Interface is enabled, panel DFHWB04 is shown. (See Figure 6 on page 59.)

```
CWBC                          CICS Web Interface                          DFHWB04

 Select one of the following. Then press Enter.

 _  1. Disable the Web Interface
    2. View or modify the Web Interface data set
    3. View or modify Web Interface status









 Current Status: Enabled


                                                  SYSID= ZAL1  APPLID= IYK2ZAL1
 PF1=Help   PF3=Exit   PF9=Messages   PF12=Return
```

Figure 6. Panel DFHWB04

The results of selecting the various options is as follows:
1. See "Disabling the CICS Web Interface" on page 61.
2. See "Updating the CICS Web Interface data set" on page 63.
3. See "Updating the CICS Web Interface status" on page 64.

Enabling the CICS Web Interface

Before enabling the CICS Web Interface, you must accept or modify some options as described in the next section.

Setting and modifying options

If you start the connection manager when the CICS Web Interface is disabled, and select option 1 on panel DFHWB01, panel DFHWB02 is shown. (See Figure 7 on page 60.)

The values displayed in the Choice column are those stored in the CICS Web Interface data set. The data set is initialized with the values shown in Figure 7 on page 60, except that the value displayed for CWBM Userid is the default CICS user ID for the CICS for MVS/ESA system in which the CICS Web Interface is operating.

You can make entries in the fields listed below. Entries may be in lowercase or uppercase. Where entries to a field are restricted (for example, YES or NO) you may enter the whole option (YES) or the minimum (Y). In the panels, the minimum entry is shown in uppercase in the Possible Options column. In the reference material in this manual, the minimum entry is given in parentheses after the full entry.

APAR PQ08890

Added unescape — RGW 23/06/1998

```
CWBC                    CICS Web Interface Enable                    DFHWB02

Overtype to Modify
                                    Choice      Possible Options

Trace                       ===>    STARTED     STArted | STOpped

Trace Level                 ===>    1           1 | 2 | A11

CWBM Userid                 ===>    CICSUSER

Analyzer Program            ===>    DFHWBADX

Automatic Enable            ===>    NO          Yes | No

Port Number                 ===>    04300       1 - 65335

Backlog                     ===>    00005       1 - 32767

Unescape                    ===>    YES         Yes | No

Enable DNS                  ===>    YES         Yes | No


Current Status: Disabled

                                                SYSID= CICS  APPLID= IYCLONCF
PF1=Help   PF3=Exit   PF4=Save   PF9=Messages   PF12=Return
```

Figure 7. Panel DFHWB02

Trace Specifies whether CICS Web Interface tracing is active. STARTED (STA) means it is active, STOPPED (STO) means it is not. The default value is STARTED.

CICS Web Interface exception trace entries are always written to CICS internal trace whatever the setting of this option. To get non-exception trace entries written, CICS trace must be started, and this option must be set to STARTED.

Trace Level
Specifies the trace level for the CICS Web Interface. The value 1 means that level 1 trace points are traced, 2 means that level 2 trace points are traced, and ALL (A) means that both level 1 and 2 are traced. The default value is 1.

CWBM Userid
Specifies the CICS user ID under which the server controller is to run. The default is the default user ID for the CICS for MVS/ESA system in which the CICS Web Interface is operating.

Analyzer Program
The name of the program that is to be used by the server controller to analyze the incoming request.

Automatic Enable
Enter YES (Y) or NO (N). If YES is stored in the CICS Web Interface data set, you can enable the CICS Web Interface by just typing CWBC; all values are defaulted from the data set, and the CICS Web Interface becomes enabled without further user input. The default value is NO.

Setting this field has an effect only when you enable the CICS Web Interface. If you use PF4 to save the values to the CICS Web Interface data

set, this value will be effective the next time you enable, unless you override it. A YES in this field in the data set may be overridden by the fast path command CWBC E A(N).

Port Number
The TCP/IP port number at which incoming HTTP requests are expected. The default port number is 80.

Backlog
The maximum number of unprocessed incoming requests to be queued by TCP/IP for MVS. If more requests are received, they are rejected. The default is 5. TCP/IP sets an upper limit defined by the SOMAXCONN parameter. See "Changes to TCP/IP for MVS" on page 16.

Unescape
Enter YES (Y) or NO (N). If this value is YES, escaped characters will be unescaped automatically by the CICS Web Interface after return from the Analyzer Program. Otherwise, the Analyzer program may specify in its return parameters whether the Web Interface must perform character unescaping. The default value is YES.

Enable DNS
If this value is YES, the CICS Web Interface is allowed to attempt to resolve hostnames via a Domain Name Server (DNS). Otherwise, DFHWBENV returns a NULL string for REMOTE_HOST (see "Reference information for the environment variables program" on page 49). The default value is YES.

Validating, saving, and activating options

After you have made your changes on panel DFHWB02, press Enter to get them validated by the connection manager. If you wish to save the new values in the CICS Web Interface data set, press PF4.

If you press Enter a second time, the CICS Web Interface becomes enabled, and panel DFHWB04 is shown.

Note: When the interface has processed a request after being enabled, it is not possible to use EXEC CICS SET PROG(DFHCNV) NEWCOPY without first disabling the interface.

Disabling the CICS Web Interface

From panel DFHWB04, select option 1, and panel DFHWB06 is shown. (See Figure 8 on page 62.)

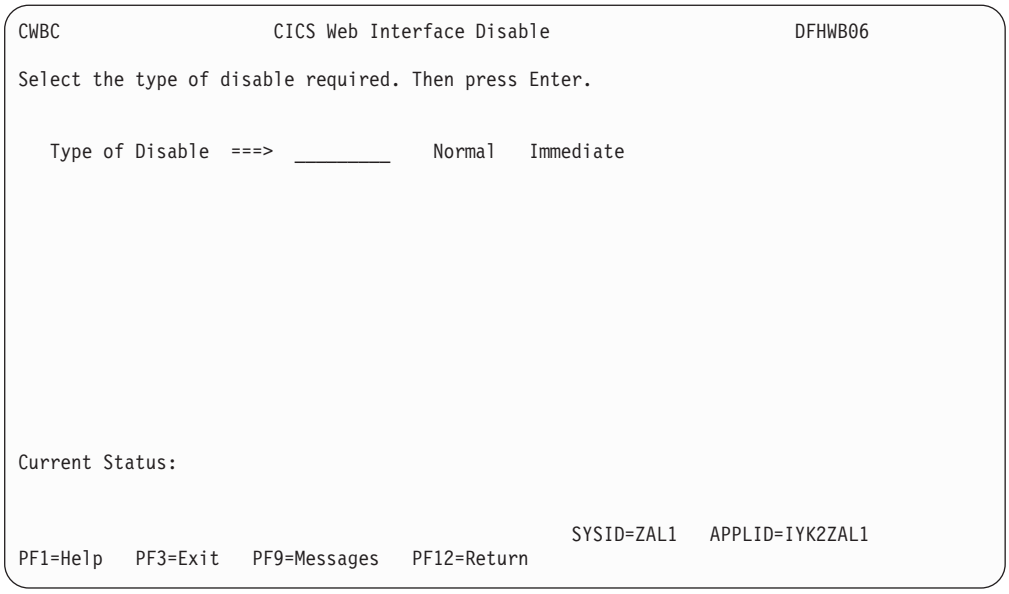

Figure 8. Panel DFHWB06

In this panel there is only one field to enter.

Type of Disable

NORMAL (N)
Normal disable processing is started. All work that has already entered the CICS Web Interface is allowed to run to completion, and replies are sent to the clients.

IMMEDIATE (I)
Immediate disable processing is started.

- Aliases not yet started do not start at all.
- CICS programs running under aliases are allowed to end, and then the alias abends. If the CICS program ends normally, and was called using DPL, the changes it makes to recoverable resources are committed. If the CICS program is a local program, the changes it makes to recoverable resources are backed out unless the CICS program takes a syncpoint with EXEC CICS SYNCPOINT.
- No replies are sent to clients, so they do not know whether the CICS program has run or not.

Pressing Enter causes the entry you have made to be validated. Pressing Enter a second time begins disable processing. The Current Status is changed to Disabling or Disabled, depending on the progress of disable processing. When disable processing is complete, pressing Enter changes the Current Status to Disabled.

The panel is displayed until you use PF3 or PF12.

On CICS for MVS/ESA normal shutdown

CICS for MVS/ESA normal shutdown starts normal disable processing for the CICS Web Interface.

On CICS for MVS/ESA immediate shutdown

On CICS for MVS/ESA immediate shutdown, all transactions are terminated. Clients are not informed of the shutdown or its effects.

Updating the CICS Web Interface data set

If you select option 2 on panel DFHWB01, or option 1 on panel DFHWB04, panel DFHWB12 is shown. (See Figure 9.)

The contents of the fields show what is currently stored in the CICS Web Interface data set.

The Current Status field in this panel might show Enabled or Disabled, depending on which panel you came from.

Once you have made your changes you should press Enter to get them validated. You can then press Enter again to update the CICS Web Interface data set with the values you have supplied. The next time you start the connection manager, the saved options are used to set up panel DFHWB02. The meanings of the options are described in "Setting and modifying options" on page 59.

APAR PQ08890

Added unescape — RGW 23/06/98

```
CWBC                CICS Web Interface Update Data Set                DFHWB12

Overtype to modify
                                    Choice       Possible Options

Trace                         ===>  STArted      STArted | STOpped

Trace Level                   ===>  1            1 | 2 | All

CWBM Userid                   ===>  CICSUSER

Analyzer Program              ===>  DFHWBADX

Automatic Enable              ===>  NO__          Yes | No

Port Number                   ===>  04300____     1 - 65335

Backlog                       ===>  00005____     1 - 32767

Unescape                      ===>  YES           Yes | No

Enable DNS                    ===>  YES           Yes | No

Current Status: Disabled

                                                      SYSID= CICS APPLID=IYCLONCF
PF1=Help   PF3=Exit   PF4=Save   PF9=Messages   PF12=Return
```

Figure 9. Panel DFHWB12

Updating the CICS Web Interface status

If you select option 2 on panel DFHWB04, panel DFHWB16 is shown. (See Figure 10.)

APAR PQ08890
Added unescape — RGW 23/06/98

```
CWBC                  CICS Web Interface Current Status              DFHWB16

Overtype to Modify
                                     Choice      Possible Options

Trace                          ===>  STArted     STArted | STOpped

Trace Level                    ===>  1           1 | 2 | A11

CWBM Userid                    ===>  CICSUSER

Analyzer Program               ===>  DFHWBADX

Port Number                    ===>  04300       1 - 65335

Backlog                        ===>  00005       1 - 32767

Unescape                       ===>  YES         Yes | No

Enable DNS                     ===>  YES         Yes | No


Current Status: Enabled

                                                  SYSID= CICS  APPLID= IYCLONCF
PF1=Help   PF3=Exit   PF4=Save   PF9=Messages   PF12=Return
```

Figure 10. Panel DFHWB16

You may change the values of the Trace, Trace Level, and Analyzer Program and unescape options. The other options are displayed for information only. Once you have made your changes you should press Enter to get them validated. You can then press Enter again to update the current status of the CICS Web Interface. The meanings of the options are described in “Setting and modifying options” on page 59.

Chapter 6. Security

This chapter is organized as follows:

- "Security for the CICS Web Interface" describes security considerations for the HTML template manger PDS, and the CICS Web Interface transactions.
- "The security sample programs" on page 66 describes the operation of the sample security analyzer, converter, and sign-on program.

Security for the CICS Web Interface

This section describes security considerations for the HTML template manager PDS, and the CICS Web Interface transactions.

Security for the HTML template manager PDS

If your CICS programs use the partitioned data set facilities of the HTML template manager described in "Reference information for the HTML template manager" on page 42, the CICS region user ID must have READ authority for the data set described in the DFHHTML DD statement.

Security for the CICS Web Interface transactions

You can specify security requirements for each of the transactions that compose the CICS Web Interface. In the following explanations:

- *authority to attach* means that the associated user must be given READ authority to the named transaction in the resource class specified by the XTRAN system initialization parameter.
- *authority to START* means that the associated user must be given READ authority to the named transaction in the resource class specified by the XPCT system initialization parameter.
- *authority to specify a user ID* means that the associated user must be given READ authority to the `userid.DFHSTART` profile in the SURROGAT resource class, if the XUSER system initialization parameter is specified as YES.

For more information, see the *CICS for MVS/ESA CICS-RACF Security Guide*.

Security for the connection manager

The connection manager transaction, CWBC, can be attached as a terminal-initiated transaction or as a started non-terminal transaction. If it is initiated from a terminal, the user signed on at that terminal must have the following authorities:

- The authority to attach the CWBC transaction itself
- The authority to START the server controller transaction, CWBM
- The authority to specify a user ID under which the CWBM transaction executes.

If CWBC is initiated by a START command, then the user associated with the transaction issuing the START must have all the above authorities, together with the authority to start the CWBC transaction.

If CWBC is initiated by a START from within a PLT program, the user ID that must have these authorities is the one specified in the PLTPIUSR system initialization parameter, or the CICS region user ID if PLTPIUSR is not specified.

Security for the server controller

The server controller is always attached as a started non-terminal transaction. It runs under the user ID specified in panel DFHWB02 (see Figure 7 on page 60), but if this is not specified, it defaults to the default CICS user ID. This user ID must have the following authorities:

- The authority to attach the server controller transaction
- The authority to START the alias transaction
- The authority to specify a user ID under which the alias transaction executes, if such a user ID is supplied in **wbra_userid** by the analyzer.

Security for the alias

The alias transaction executes as a non-terminal CICS transaction. Its name is user-specified. If you use the default analyzer described in "The default analyzer" on page 21, the transaction name is the second "index level" in the absolute path specified by the client, and is usually CWBA.

The alias transaction executes under the user ID specified in **wbra_userid**, if it is specified by the analyzer, otherwise it executes under the same user ID as the server controller. This user ID must have the following authorities:

- The authority to attach the alias transaction
- The authority to access any CICS resources used by the alias transaction, if it is defined with the RESSEC(YES) option
- The authority to access any CICS system programming commands used by the alias transaction, if it is defined with the CMDSEC(YES) option

If the alias transaction has insufficient security access to execute, a security violation message is sent to the CICS log, and an error code of 403 (Forbidden) is sent to the browser.

The security sample programs

If you want a series of web transactions to be executed under a user ID that is specified by the web client (the end-user), you can use the following security sample programs to help you.

APAR PQ08890
Added unescape — RGW 23/06/98

- DFH$WBSA is the security analyzer sample program. If you use this program, you must specify its name as the Analyzer Program name and set Unescape to YES in panel DFHWB02 when you enable the interface. (See Figure 7 on page 60).
- DFH$WBSC is the security converter sample program.
- DFH$WBSN is the sign-on sample program.

These programs use the state management sample program, DFH$WBST.

A typical sequence of interactions between a user and the CICS Web Interface might be as follows:

1. The end user sends an HTTP request in which the URL has no query string.
2. The security analyzer checks the URL for a converter name, alias name, program name, and query string. As there is no query string, it sets its

outputs so that the converter is the security converter sample program DFH$WBSC, while the alias and CICS program are the ones requested in the URL. The user token output is zeroes.

3.

APAR PQ07658
Changed program name ″SIGNON″ to ″DFHWBSN— RGW 22/06/1998

The **Decode** function of the security converter, finding a zero user token, calls the Create function of the state management sample program to assign a token. It saves the token in its user token output. It uses the Store function of the state management program to save the original URL. It changes the CICS program name to DFHWBSN, which is an alias of the sign-on sample program, DFH$WBSN.

4. The sign-on program builds an HTML form asking for a user ID and a password. The form specifies an HTML ACTION that generates a URL. The generated URL causes the sign-on program to be invoked again, but with the state management token as its query string.
5. The **Encode** function of the security converter builds the HTTP response.
6. The user gets the form, fills in the user ID and the password, and sends it back.
7. The security analyzer finds a query string. It uses the Retrieve function of the state management program to validate the token. As the token is not yet associated with a valid user ID, it sets its outputs so that the converter name is the security converter. The state management token is passed as the user token.
8. The sign-on program extracts the user ID and password from the form, and uses EXEC CICS VERIFY PASSWORD to validate the user ID. It uses the Store function of the state management program to associate the validated user ID with the token.
9. The **Encode** function of the security converter builds the HTTP response, and adds a redirection (HTTP response 302) to it, incorporating the original URL.
10. The web browser receives the redirected URL, and sends a request for the original program with the token that identifies the validated user ID.
11. The security analyzer finds that the query string is a valid user token associated with a user ID, so the original request proceeds.

Once the user token has been established as the key to the authenticated user ID, it is the responsibility of the CICS program, or the converter that builds the HTTP response, to ensure that any URLs that are generated to continue the conversation with the client contain the conversation token as query string. This ensures that subsequent programs in the conversation execute under the specified user ID. Since the CICS program is already running with the correct conversation token as its query string, it can extract its value by using the environment variable program to obtain the value of the query string. If necessary, the correct value for the conversation token can be substituted into HTML templates by using the symbol `&QUERY_STRING;`, provided that the environment variable string has first been loaded into the symbol table in the template manager's page environment.

Chapter 7. Problem determination

This chapter helps you debug problems in CICS Web Interface user-replaceable programs, the IBM-supplied parts of the CICS Web Interface, and in the system setup of the CICS Web Interface. If you suspect you have a problem in another part of CICS, refer to the *CICS for MVS/ESA Problem Determination Guide*.

The formats of messages and trace outputs in the CICS Web Interface are also described.

Diagnostic information is designed to provide first failure data capture, so that if an error occurs, enough information about the error is available directly without the need to reproduce the error situation. The information is presented in the following forms:

Messages. The CICS Web Interface provides CICS messages with the prefix DFHWB, and these are listed in "Appendix B. Messages and Codes" on page 93.

Trace. The CICS Web Interface outputs system trace entries containing all the important information required for problem diagnosis. The CICS Web Interface trace points are listed in "CICS Web Interface trace information" on page 72.

Dump. Dump formatting is provided for data areas relating to the CICS Web Interface.

Abend codes. Transaction abend codes are standard four-character names. The abend codes output by the CICS Web Interface are listed in "Appendix B. Messages and Codes" on page 93.

This chapter is organized as follows:

- "Recovery procedures" on page 70 describes how the CICS Web Interface copes with software errors.
- "Product design considerations" on page 70 describes aspects of the design of the CICS Web Interface that you need to know for problem determination.
- "Troubleshooting" on page 70 describes a method of analyzing problems in the CICS Web Interface.
- "Using messages and codes" on page 71 describes how to find information about CICS Web Interface messages and abend codes.
- "CICS Web Interface trace information" on page 72 describes the CICS Web Interface trace information, and gives a list of trace points.
- "Dump and trace formatting" on page 82 describes how to control the formatting of dumps and trace entries.
- "Debugging the user-replaceable programs" on page 82 gives hints about debugging user-replaceable programs.

This chapter contains Diagnosis, Modification, or Tuning Information.

Recovery procedures

Software errors within the server controller may cause it to perform an immediate disable (if this is not prevented by the nature of the error). After an immediate disable of the CICS Web Interface, CICS continues to run.

The CICS Web Interface is not included in CICS recovery. If CICS abends and is then restarted, you must enable the CICS Web Interface as described in “Enabling the CICS Web Interface” on page 59.

If TCP/IP for MVS abends, the CICS Web Interface enters immediate disable processing, but CICS continues to run.

The abending of an alias transaction might cause changes to recoverable resources to be backed out.

Product design considerations

There are some aspects of product design that you need to be aware of for problem determination:

- Use of different MVS TCBs for processing client requests
- Use of a task-related user exit

The server controller interacts with TCP/IP for MVS, and uses the MVS WAIT service. This processing must be done under CICS's RP TCB, rather than under the QR TCB, which is used for most other processing.

The CICS Web Interface uses a task-related user exit to anchor shared storage, and to improve its response to CICS shutdown processing.

Troubleshooting

This section provides some hints on troubleshooting. It follows the general outline:

1. Define the problem.
2. Obtain information (documentation) on the problem.
3. Work out where in the CICS Web Interface the problem is happening.

Defining the problem

When you have a problem, first try to define the circumstances that gave rise to it. If you need to report the problem to the IBM software support center, this information is useful to the personnel there.

1. What is the system configuration?
   a. CICS for MVS/ESA release
   b. TCP/IP for MVS release
   c. Language Environment/370 release
   d. MVS release
2. What operating options are in use?
3. When did the problem first occur?
4. What were you trying to accomplish at the time the problem occurred?
5. What changes were made to the system before the occurrence of the problem?
   a. To the MVS system
   b. To the CICS Web Interface
   c. To the CICS program being called by the client
   d. To the converter being used in the call

e. To the analyzer being used to interpret client requests
f. To the client
g. To the CICS for MVS/ESA
h. To TCP/IP for MVS

6. What is the problem?
   a. Incorrect output
   b. Hang/Wait: Use CEMT INQUIRE to display details of the transaction.
   c. Loop: Use CEMT INQUIRE to display the details of the transaction.
   d. Abend in a user-replaceable program
   e. Abend in a CICS program
   f. Abend in the IBM-supplied part of the CICS Web Interface
   g. Performance problem
   h. Storage violation
   i. Logic Error
7. At what point in the processing did the problem occur? (Use Figure 3 on page 7.)
8. What was the state of TCP/IP for MVS? (Try the **rpcinfo** command.)

Documentation about the problem

To investigate most problems, you must look at the dumps, traces, and logs provided with MVS and CICS.

1. System dump: This contains the CICS internal trace.
2. CICS auxiliary trace, if enabled
3. TCP/IP for MVS trace
4. GTF trace, if enabled
5. Console log
6. CSMT log
7. CWBO log
8. CICS job log

To identify which are likely to be useful for your problem, try to work out the area of the CICS Web Interface giving rise to the problem, and read the relevant section in the rest of this chapter.

Using messages and codes

CICS Web Interface messages have identifiers of the form DFHWBnnnn, where nnnn are four numeric characters. These numbers indicate which of the CICS Web Interface components generated the message, as shown in "Appendix B. Messages and Codes" on page 93.

Messages are generated for end-users by the connection manager. They are sent to the CICS Web Interface message transient data queue CWBO, or the terminal user, or both, depending on the event that is being reported. If you define CWBO as an indirect destination for CSMT, the messages appear in CSMT. Some messages are sent to the console.

When the CICS Web Interface issues a message as a result of an error, it also makes an exception trace entry. The CICS Web Interface also generates information messages, for instance during enable processing and disable processing.

CICS Web Interface messages are supplied in English, Japanese, and Chinese. The CICS message editing utility can be used to translate them into other languages supported by CICS, as described in the *CICS for MVS/ESA Operations and Utilities Guide*.

The CICS Web Interface abend codes are listed in "Appendix B. Messages and Codes" on page 93.

CICS Web Interface trace information

The CICS Web Interface outputs CICS system trace, which is formatted using software supplied as part of the CICS Web Interface.

Exception trace entries produced by the CICS Web Interface are written to CICS internal trace even when the Trace operating option is set to NO. See "Setting and modifying options" on page 59 for information about the Trace option.

If selected, level 2 trace gives a full trace of the data being transmitted between the client and the CICS program. CICS trace output is described in the *CICS for MVS/ESA Problem Determination Guide*.

CICS Web Interface trace points

Table 7 lists all the trace points in the CICS Web Interface in numeric order. For each trace point, the following information is given:

Point ID
CICS Web Interface trace points are hexadecimal numbers in the range 0000 to FFFF with a prefix FT.

Module
Name of the module that produces the trace entry.

Lvl Trace point levels are 1, 2, or exception. Exception trace entries are output regardless of the trace level setting.

Type The circumstances that gave rise to the trace entry.

Data The data that is traced in the trace entry.

APAR PQ11796
Changes to trace point 9F28 and new trace point 9F2D

Table 7. CICS Web Interface trace points

Point ID	Module	Lvl	Type	Data
FT 9E00	DFHWBC01	1	Entry	None
FT 9E01	DFHWBC01	1	Exit	None
FT 9E02	DFHWBC01	Exc	Invalid GWA length	**1** WBCOM **2** HTTP global work area
FT 9E03	DFHWBC01	Exc	EXTRACT error	**1** WBCOM **2** EIB
FT 9E04	DFHWBC01	Exc	CWBM running	**1** WBCOM
FT 9E05	DFHWBC01	Exc	TRUE enabled without CWBM	**1** WBCOM

Table 7. CICS Web Interface trace points (continued)

Point ID	Module	Lvl	Type	Data
FT 9E06	DFHWBC01	Exc	ASSIGN error	**1** WBCOM **2** EIB
FT 9E07	DFHWBC01	Exc	Invalid terminal	**1** WBCOM
FT 9E08	DFHWBC01	Exc	REGISTER error	**1** DUFT parameters
FT 9E09	DFHWBC01	Exc	GETMAIN buffer error	**1** EIB
FT 9E0A	DFHWBC01	Exc	RECEIVE error	**1** EIB
FT 9E0B	DFHWBC01	Exc	RETRIEVE error	**1** EIB
FT 9E0C	DFHWBC01	Exc	Invalid STARTCODE	**1** EIB
FT 9E30	DFHWBC42	1	Entry	**1** WBCOM
FT 9E31	DFHWBC42	1	Exit	**1** WBCOM
FT 9E32	DFHWBC42	Exc	INQUIRE SECURITYMGR error	**1** WBCOM
FT 9E33	DFHWBC42	Exc	Update ENQ error	**1** WBCOM **2** HTTP global work area **3** EIB
FT 9E34	DFHWBC42	Exc	Update DEQ error	**1** WBCOM **2** HTTP global work area **3** EIB
FT 9E35	DFHWBC42	Exc	ENABLE EXIT error	**1** EIB **2** WBCOM
FT 9E36	DFHWBC42	Exc	EXTRACT EXIT error	**1** EIB **2** WBCOM **3** HTTP global work area
FT 9E37	DFHWBC42	Exc	Invalid GWA status	**1** HTTP global work area
FT 9E38	DFHWBC42	Exc	START CWBM error	**1** EIB **2** WBCOM **3** HTTP global work area
FT 9E39	DFHWBC42	Exc	DISABLE TRUE error	**1** EIB **2** WBCOM

Table 7. CICS Web Interface trace points (continued)

Point ID	Module	Lvl	Type	Data
FT 9E3A	DFHWBC42	Exc	UPDATE_FEATURE error	**1** DUFT parameters
FT 9E60	DFHWBC09	1	Entry	**1** WBCOM
FT 9E61	DFHWBC09	1	Exit	**1** WBCOM
FT 9E62	DFHWBC09	Exc	READ error	**1** WBCOM **2** EIB
FT 9E63	DFHWBC09	Exc	WRITE error	**1** WBCOM **2** Data record
FT 9E64	DFHWBC09	Exc	READ update error	**1** WBCOM **2** Data record
FT 9E65	DFHWBC09	Exc	WRITE update error	**1** WBCOM **2** Data record
FT 9E90	DFHWBC0B	1	Message issue entry	**1** WBCOM
FT 9E91	DFHWBC0B	1	Message issue exit	**1** WBCOM
FT 9E92	DFHWBC0B	Exc	WRITEQ error	**1** EIB **2** WBCOM
FT 9E93	DFHWBC0B	Exc	SEND message error	**1** EIB **2** WBCOM
FT 9E94	DFHWBC0B	1	Display errors entry	**1** WBCOM
FT 9E95	DFHWBC0B	1	Display errors exit	**1** WBCOM
FT 9E96	DFHWBC0B	1	Help panels entry	**1** WBCOM
FT 9E97	DFHWBC0B	1	Help panels exit	**1** WBCOM
FT 9E98	DFHWBC0B	Exc	EXTRACT EXIT error	**1** WBCOM **2** EIB
FT 9E99	DFHWBC0B	Exc	Invalid GWA length	**1** WBCOM **2** EIB
FT 9E9A	DFHWBC0B	Exc	Invalid GWA	**1** WBCOM **2** EIB
FT 9E9B	DFHWBC0B	Exc	Message GETMAIN error	**1** EIB **2** WBCOM

Table 7. CICS Web Interface trace points (continued)

Point ID	Module	Lvl	Type	Data
FT 9EC0	DFHWBC04	1	Entry	**1** WBCOM
FT 9EC1	DFHWBC04	1	Exit	**1** WBCOM
FT 9EC2	DFHWBC04	1	Update GWA entry	**1** WBCOM
FT 9EC3	DFHWBC04	1	Update GWA exit	**1** WBCOM
FT 9EC4	DFHWBC04	Exc	Update DEQ error	**1** WBCOM **2** HTTP global work area **3** EIB
FT 9EC5	DFHWBC04	Exc	Update ENQ error	**1** WBCOM **2** HTTP global work area **3** EIB
FT 9EF0	DFHWBC03	1	Entry	**1** WBCOM
FT 9EF1	DFHWBC03	1	Exit	**1** WBCOM
FT 9EF2	DFHWBC03	Exc	ENQ error	**1** WBCOM **2** EIBRESP
FT 9EF3	DFHWBC03	Exc	DEQ error	**1** WBCOM **2** EIBRESP
FT 9F01	DFHWBWB	Exc	Invalid format	**1** WBWB parameters
FT 9F02	DFHWBWB	Exc	Invalid function	**1** WBWB parameters
FT 9F03	DFHWBWB	1	Entry	**1** WBWB parameters
FT 9F04	DFHWBWB	1	Exit	**1** WBWB parameters
FT 9F05	DFHWBWB	Exc	Recovery	**1** WB parameters **2** Kernel error data
FT 9F06	DFHWBWB	1	Call	**1** C parameters **2** HTTP parameters
FT 9F07	DFHWBWB	1	Return	**1** C parameters **2** HTTP parameters
FT 9F08	DFHWBWB	Exc	Invalid GWA	Not traced

Table 7. CICS Web Interface trace points (continued)

Point ID	Module	Lvl	Type	Data
FT 9F09	DFHWBWB	Exc	Initialization error	**1** WBWB parameters **2** HTTP parameters
FT 9F0A	DFHWBWB	Exc	No storage	**1** WBWB parameters
FT 9F0B	DFHWBWB	Exc	Socket error	**1** WBWB parameters **2** HTTP parameters
FT 9F0C	DFHWBWB	Exc	GETHOSTID error	**1** WBWB parameters **2** HTTP parameters
FT 9F0D	DFHWBWB	Exc	Port in use	**1** WBWB parameters **2** HTTP parameters
FT 9F0E	DFHWBWB	Exc	Port not available	**1** WBWB parameters **2** HTTP parameters
FT 9F0F	DFHWBWB	Exc	BIND error	**1** WBWB parameters **2** HTTP parameters
FT 9F10	DFHWBWB	Exc	IOCTL error	**1** WBWB parameters **2** HTTP parameters
FT 9F11	DFHWBWB	Exc	Logic error	**1** Trace point **2** WBWB parameters
FT 9F12	DFHWBWB	Exc	LISTEN error	**1** WBWB parameters **2** HTTP parameters
FT 9F13	DFHWBWB	Exc	DELETE_LOCK error	**1** WBWB parameters
FT 9F14	DFHWBWB	Exc	Duplicate lock error	**1** WBWB parameters
FT 9F15	DFHWBWB	Exc	LOCK error	**1** WBWB parameters
FT 9F16	DFHWBWB	Exc	Bad parameter list	**1** WBWB parameters **2** HTTP parameters

Table 7. CICS Web Interface trace points (continued)

Point ID	Module	Lvl	Type	Data
FT 9F19	DFHWBWB	Exc	Handle not found	**1** WBWB parameters
FT 9F1A	DFHWBWB	Exc	SEND error	**1** WBWB parameters **2** HTTP parameters
FT 9F1C	DFHWBWB	Exc	CLOSE error	**1** WBWB parameters **2** HTTP parameters
FT 9F1D	DFHWBWB	Exc	SELECT error	**1** WBWB parameters **2** HTTP parameters
FT 9F1E	DFHWBWB	Exc	ACCEPT error	**1** WBWB parameters **2** HTTP parameters
FT 9F20	DFHWBWB	Exc	Invalid socket	**1** WBWB parameters **2** HTTP parameters
FT 9F21	DFHWBWB	Exc	Connection lost	**1** WBWB parameters **2** HTTP parameters
FT 9F22	DFHWBWB	Exc	RECEIVE storage error	**1** WBWB parameters
FT 9F23	DFHWBWB	Exc	Blocking state error	**1** WBWB parameters **2** HTTP parameters
FT 9F24	DFHWBWB	Exc	SEND invalid socket	**1** WBWB parameters **2** HTTP parameters
FT 9F25	DFHWBWB	Exc	SEND storage error	**1** WBWB parameters
FT 9F27	DFHWBWB	Exc	GETHOSTNAME error	**1** WBWB parameters **2** HTTP parameters
FT 9F28	DFHWBWB	Exc	Unexpected RECEIVE error	**1** WBWB parameters **2** HTTP parameters **3** HTTP caller global area

Table 7. CICS Web Interface trace points (continued)

Point ID	Module	Lvl	Type	Data
FT 9F29	DFHWBWB	1	No data received	**1** WBWB parameters **2** HTTP parameters **3** HTTP caller global area
FT 9F2A	DFHWBWB	1	Not HTTP request	**1** WBWB parameters
FT 9F2D	DFHWBWB	Exc	GETHOSTBYADDR error	**1** WBWB parameters **2** HTTP parameters **3** HTTP caller global area
FT 9F2E	DFHWBWB	Exc	ACCEPT not possible	**1** WBWB parameters **2** HTTP parameters **3** DFHWBCGC
FT 9F33	FT 9F33	Exc	Runaway detected on SEND	**1** WBWB parameters **2** HTTP parameters **3** DFHWBCGC
FT 9F34	DFHWBWB	Exc	MAXDESC failed	**1** WBWB parameters **2** HTTP parameters **3** HTTP caller global area
FT 9F35	DFHWBWB	Exc	SEND SELECT error	**1** WBWB parameters **2** HTTP parameters **3** HTTP caller global area
FT 9F36	DFHWBWB	Exc	TCP/IP no longer active	**1** WBWB parameters **2** HTTP parameters **3** HTTP caller global area
FT A000	DFHWBM	1	Entry	None
FT A001	DFHWBM	1	Exit	None
FT A002	DFHWBM	2	Input in client code page	**1** Input data in ASCII
FT A003	DFHWBM	1	Input HTTP data	**1** First 256 bytes
FT A003	DFHWBM	2	Input HTTP data	**1** Remaining data

Table 7. CICS Web Interface trace points (continued)

Point ID	Module	Lvl	Type	Data
FT A004	DFHWBM	1	Input user data	**1** First 256 bytes
FT A004	DFHWBM	2	Input user data	**1** Remaining data
FT A005	DFHWBM	1	Analyzer request	**1** Analyzer parameters
FT A006	DFHWBM	1	Analyzer response	**1** Analyzer parameters
FT A007	DFHWBM	Exc	Disable TRUE error	**1** EIB
FT A008	DFHWBM	Exc	Handle analyzer abend	**1** Analyzer parameters **2** Global work area **3** Web request block
FT A009	DFHWBM	Exc	CICS abend (client)	**1** Global work area **2** Web request block
FT A00A	DFHWBM	Exc	CICS abend (no client)	**1** Global work area
FT A00B	DFHWBM	Exc	EXTRACT EXIT error	**1** EIB
FT A00C	DFHWBM	Exc	Corrupt GWA	**1** Global work area
FT A00D	DFHWBM	Exc	Select error	**1** WBWB parameters **2** Global work area
FT A00E	DFHWBM	Exc	Analyzer LINK error	**1** EIB **2** Global work area **3** Analyzer parameters **4** Web request block
FT A00F	DFHWBM	Exc	Analyzer return error	**1** Analyzer parameters **2** Global work area **3** Web request block
FT A010	DFHWBM	Exc	GETMAIN error	**1** EIB **2** Global work area
FT A011	DFHWBM	Exc	WAIT MVS error	**1** DSSR parameters **2** Global work area
FT A013	DFHWBM	Exc	START error	**1** EIB **2** Global work area **3** Web request block

Table 7. CICS Web Interface trace points (continued)

Point ID	Module	Lvl	Type	Data	
FT A014	DFHWBM	Exc	CHANGE MODE error	**1** **2**	DSAT parameters Global work area
FT A015	DFHWBM	Exc	Free WBWB error	**1**	EIB
FT A016	DFHWBM	Exc	Activate TCB error	**1**	DSIT parameters
FT A017	DFHWBM	Exc	Add subpool error	**1**	SMAD parameters
FT A018	DFHWBM	Exc	Load WBWB error	**1**	EIB
FT A019	DFHWBM	Exc	Initialize HTTP error	**1** **2**	WBWB parameters Global work area
FT A100	DFHWBA	1	Entry	**1**	None
FT A101	DFHWBA	1	Exit	**1**	None
FT A102	DFHWBA	Exc	LINK error	**1**	EIB WRB
FT A103	DFHWBA	Exc	STARTCODE error	**1**	EIB
FT A104	DFHWBA	Exc	RETRIEVE data error	**1**	EIB
FT A107	DFHWBA	Exc	FREEMAIN output error	**1**	EIB
FT A109	DFHWBA	Exc	Global work are error	**1** **2** **3**	Enable time Global work area WRB
FT A10B	DFHWBA	1	Server output data	**1**	First 256 bytes
FT A10C	DFHWBA	2	Output data continued	**1**	Remaining data
FT A10D	DFHWBA	Exc	CHANGE_MODE	**1**	WRB
FT A10E	DFHWBA	Exc	Return error	**1**	WRB
FT A10F	DFHWBA	Exc	Main abend	**1**	WRB
FT A110	DFHWBA1	1	DECODE request	**1**	**Decode** parameters
FT A111	DFHWBA1	1	DECODE response	**1**	**Decode** parameters

Table 7. CICS Web Interface trace points (continued)

Point ID	Module	Lvl	Type	Data
FT A112	DFHWBA1	1	ENCODE request	**1** **Encode** parameters
FT A113	DFHWBA1	1	ENCODE response	**1** **Encode** parameters
FT A114	DFHWBA1	Exc	LINK terminal error	**1** Alias parameters **2** EIB
FT A115	DFHWBA1	Exc	LINK error	**1** Alias parameters **2** EIB
FT A116	DFHWBA1	Exc	Main abend	**1** Alias parameters
FT A117	DFHWBA1	Exc	Return error	**1** Alias parameters
FT A118	DFHWBA1	Exc	CORRUPT_CLIENT_DATA	**1** Alias parameters
FT A119	DFHWBA1	Exc	Security error	**1** Alias parameters
FT A11A	DFHWBA1	Exc	EXCEPTION error	**1** Alias parameters
FT A11B	DFHWBA1	Exc	INVALID response	**1** Alias parameters
FT A11C	DFHWBA1	Exc	DISASTER response	**1** Alias parameters
FT A11D	DFHWBA1	Exc	Error response	**1** Alias parameters
FT A11F	DFHWBA1	Exc	Parameter error	**1** New eyecatcher **2** Old eyecatcher
FT A120	DFHWBA1	1	Entry	None
FT A121	DFHWBA1	1	Exit	None
FT A200	DFHWBTL	1	Entry	**1** DFHWBTL parameters
FT A201	DFHWBTL	1	Exit	**1** DFHWBTL parameters
FT A300	DFHWBENV	1	Entry	None
FT A301	DFHWBENV	1	Exit	None
FT A302	DFHWBENV	Exc	Invalid task number	None
FT A303	DFHWBENV	Exc	WRB not found	None
FT A304	DFHWBENV	Exc	GWA not found	None

Dump and trace formatting

To select the level of dump formatting printed for the CICS Web Interface, you change the CICS VERBEXIT in the IPCS control statement for dump formatting as follows:

```
IPCS VERBEXIT CICS410 FT=0|1|2|3,TR=1|2
```

The parameters have these meanings:

FT=0 Suppress system dumps for the CICS Web Interface, and for CICS features.

FT=1 Produce a system dump summary listing for the CICS Web Interface, and for CICS features.

FT=2 Produce a system dump for the CICS Web Interface, and for CICS features.

FT=3 Produce a system dump summary listing and a system dump for the CICS Web Interface, and for CICS features.

TR=1 Produce an abbreviated trace.

TR=2 Produce a full trace.

Each trace entry for the CICS Web Interface has a comment "`CICS WEB`" to distinguish it from other trace points with the FT prefix.

CICS Web Interface output in the formatted dump consists of the major CICS Web Interface control blocks, with interpretation of some of the fields. The CICS Web Interface output can be found in the IPCS output by searching for "`===WB`", and is under the heading "`CICS Web Interface`"

Debugging the user-replaceable programs

The user-replaceable programs are
- The analyzer
- The converters

Using EDF

You cannot use EDF with the analyzer, but you can use it to debug the converter. If you want to use EDF, you must:

- Write the analyzer so that it specifies a terminal for EDF in the **wbra_alias_termid** output field. The chosen terminal must be a local terminal that is logged on or predefined.
- Define CEDF(YES) in the program definition of the converter.
- Define EDF as a translator option when the converter is translated.
- Enter CEDF ON at the chosen terminal.

Using trace entries

Diagnostic information may be output to the CICS trace by the use of the EXEC CICS ENTER TRACENUM command. The amount of trace information and the information contained within trace entries is at your discretion. See the *CICS for MVS/ESA Application Programming Reference* for more information about this command.

Writing messages

Diagnostic messages may be output by using EXEC CICS WRITEQ TD. Message information content, message format, frequency, and destination are at your discretion.

Abends

You are recommended to use EXEC CICS HANDLE ABEND to trap abends. You should collect the diagnostic information you need by tracing, and so on, and then return a URP_DISASTER response.

Appendix A. The business logic interface

This appendix:

- "A two-tier programming model" explains the separation of presentation logic from business logic.
- "What is the business logic interface?" on page 86 describes the business logic interface in CICS application design, and describes how it might be exploited.
- "Reference information" on page 87 provides reference information for callers of the business logic interface.

A two-tier programming model

A simple CICS application to accept data from an end user, update a record in a file, and send a response back to the end user is illustrated in Figure 11. The transaction that runs this program is the second in a pseudoconversation. The first transaction has sent a map to the end user's terminal, and this transaction reads the data with EXEC CICS RECEIVE MAP, updates the record in the file, and sends the response with EXEC CICS SEND MAP. The RECEIVE MAP and SEND MAP commands are part of the transaction's *presentation logic*, while the READ UPDATE and REWRITE commands are part of its *business logic*.

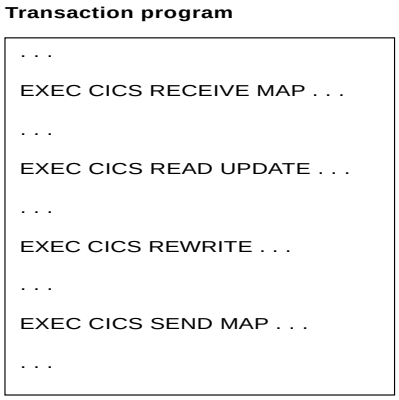

Figure 11. CICS functions in a single transaction program

A sound principle of modular programming in CICS application design is to separate the presentation logic from the business logic, and use a communication area and the EXEC CICS LINK command to make them into a single transaction. Figure 12 on page 86 illustrates this approach to application design.

Once the business logic of a transaction has been isolated from the presentation logic and given a communication area interface, it is available for reuse with different presentation methods. The CICS Web Interface encourages the use of the two-tier model when the presentation logic is HTTP-based.

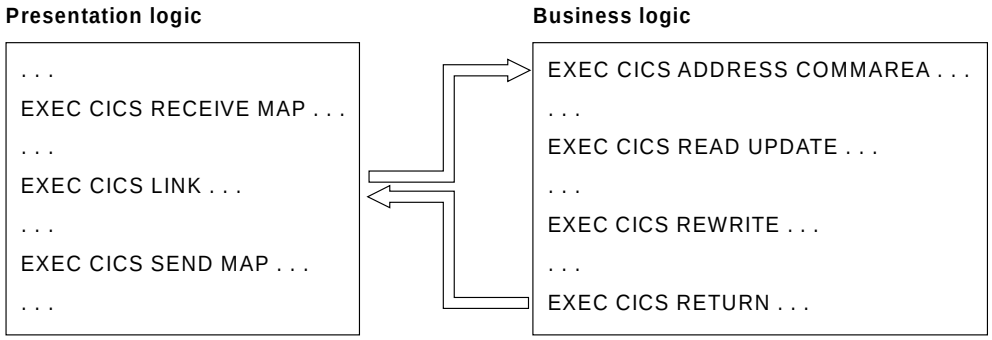

Figure 12. Separation of presentation logic from business logic

What is the business logic interface?

The business logic interface is a facility that allows you to associate business logic and presentation logic is a standard way. The presentation logic is placed in a converter. The inbound presentation logic is in the **Decode** function, and the outbound logic is in the **Encode** function.

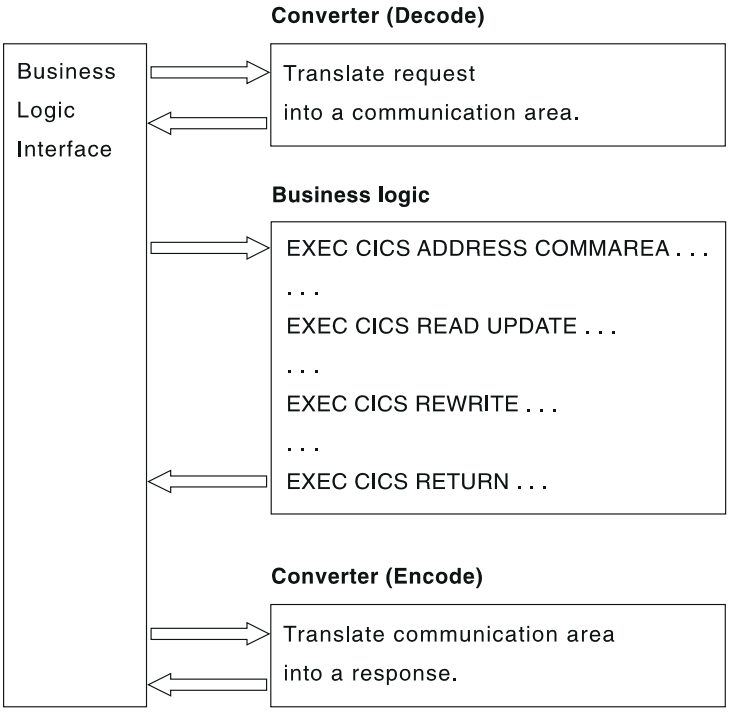

Figure 13. Using the business logic interface

Figure 13 shows how the business logic interface program DFHWBA1 supports the isolation of business logic by calling **Decode** and **Encode** functions of a converter to provide the presentation logic appropriate to the requestor. The requestor supplies a communication area containing:

- The name of the converter whose **Decode** and **Encode** functions are to be used
- The name of the CICS program that contains the business logic

- The data that **Decode** is to use to build the communication area for the CICS program.

The business logic interface can be used as follows:

- Existing CICS programs on any CICS system can use EXEC CICS LINK to call DFHWBA1.
- Users of the EXCI can call DFHWBA1.
- CICS clients on remote workstations can use the ECI to call DFHWBA1.

Reference information

This section contains reference information for the business logic interface.

Business logic interface

Summary of parameters

The names of the parameters and constants, translated into appropriate forms for the different programming languages supported, are defined in files supplied as part of the CICS Web Interface. The files for the various languages are as follows:

Language	File
Assembler	DFHWBA1D
C	DFHWBA1H
COBOL	DFHWBA1O
PL/I	DFHWBA1L

These files give language-specific information about the data types of the fields in the communication area.

In the following table, the names of the parameters are given in abbreviated form: each name in the table must be prefixed with **wba1_** to give the name of the parameter.

Table 8. Parameters for the business logic interface

Input wba1_	Inout wba1_	Output wba1_
blip_eyecatcher **client_address** **client_address_string** **converter_program_name** **data_ptr** **header_length** **header_offset** **http_version_offset** **input_data_length** **method_length** **method_offset** **resource_length** **resource_offset** **server_program_name** **user_data_length** **user_data_offset** **version_length**	**user_token**	**outdata_ptr** **response**

Function

The business logic interface allows callers to specify what presentation logic is to be executed before and after a CICS program. It has two modes of operation:

- Pointer mode: the input data for data for **Decode** is in storage allocated separately from the communication area for the business logic interface. The communication area contains a pointer (**wba1_data_ptr**) to the input data for **Decode**. When the call to the business logic interface ends, the output from **Encode** is in storage allocated separately from the communication area for the business logic interface, and the communication area contains a pointer (**wba1_outdata_ptr**) to the output from **Encode**.
- Offset mode: the input data for data for **Decode** is part of the communication area for the business logic interface. The communication area contains the offset (**wba1_data_ptr**) of the input data for **Decode**. When the call to the business

logic interface ends, the output from **Encode** is part of the communication area for the business logic interface, and the communication area contains the offset (**wba1_outdata_ptr**) of the output from **Encode**.

The caller of the business logic interface uses **wba1_blip_eyecatcher** to indicate which mode of operation is to be used.

For information about writing a converter for the business logic interface, see "Chapter 3. Writing user-replaceable programs for the CICS Web Interface" on page 19.

Note: The business logic interface does not handle the response codes and reason codes produced by the converter in the manner described in "Chapter 3. Writing user-replaceable programs for the CICS Web Interface" on page 19, but as described in "Responses" on page 91 under responses 400, 500, and 501.

Parameters

Before inserting the inputs into the communication area, you must clear it to binary zeros.

wba1_blip_eyecatcher
(Input only)

A string of length 8. In pointer mode, it must contain "**BLIP**". In offset mode, it must contain "**BLI0**"

wba1_client_address
(Input only)

The 32-bit IP address of the client.

wba1_client_address_string
(Input only)

A 15-byte string containing the client IP address in dotted decimal format.

wba1_converter_program_name
(Input only)

An 8-byte string containing the name of the converter. If you put spaces or nulls in this field, no converter is called.

wba1_data_ptr
(Input only)

In pointer mode, a pointer to the request. In offset mode, the offset of the request in the communication area.

wba1_header_length
(Input only)

The length in bytes of the header information in the HTTP request. If the request is not an HTTP request, do not set this field.

wba1_header_offset
(Input only)

The offset of the first header in the HTTP request. If the request is not an HTTP request, do not set this field.

wba1_http_version_length
(Input only)

The length in bytes of the version in the HTTP request. If the request is not an HTTP request, do not set this field.

wba1_http_version_offset
(Input only)

The offset of the version in the HTTP request. If the request is not an HTTP request, do not set this field.

wba1_input_data_length
(Input only)

The length in bytes of the request.

wba1_method_length
(Input only)

The length in bytes of the method in the HTTP request. If the request is not an HTTP request, do not set this field.

wba1_method_offset
(Input only)

The offset of the method in the HTTP request. If the request is not an HTTP request, do not set this field.

wba1_outdata_ptr
(Output only)

In pointer mode, a pointer to the output from **Encode**. In offset mode, the offset of the output from **Encode** in the communication area.

wba1_resource_length
(Input only)

The length in bytes of the absolute path in the HTTP request. If the request is not an HTTP request, do not set this field.

wba1_resource_offset
(Input only)

The offset of the absolute path in the HTTP request. If the request is not an HTTP request, do not set this field.

wba1_response
(Output only)

A response code—see “Response codes” below.

wba1_server_program_name
(Input only)

An 8-byte string identifying the program to be called to service the request.

wba1_user_data_length
(Input only)

The length of the user data in the HTTP request. If the analyzer modified this value it is visible here. If the request is not an HTTP request, do not set this field.

wba1_user_data_offset
(Input only)

The offset of the user data in the HTTP request. If the request is not an HTTP request, do not set this field.

wba1_user_token
(Input and output)

A 64-bit token. On input, it is the input to **Decode** in **decode_user_token**. On output, it is the output from **Decode** in **decode_user_token**.

Responses

One of the following values is returned in **wba1_response**. These values correspond to the intended HTTP responses to be sent to an HTTP client.

400 One of the converter functions returned a URP_EXCEPTION response with a reason URP_CORRUPT_CLIENT_DATA. The business logic interface writes an exception trace entry (trace point A115) and issues a message (DFHWB0120).

403 The EXEC CICS LINK to the program specified in **wba1_server_program_name** received a NOTAUTH response. The business logic interface writes an exception trace entry (trace point A115) and issues a message (DFHWB0120).

404 The EXEC CICS LINK to the program specified in **wba1_server_program_name** received a PGMIDERR response. The business logic interface writes an exception trace entry (trace point A115) and issues a message (DFHWB0120).

500 One of the following occurred:

- The business logic interface detected an abend. A message that depends on the program that abended is issued. For the program specified in **wba1_server_program_name**, the message is DFHWB0125. For the **Encode** function of the converter, the message is DFHWB0126. For the **Decode** function of the converter, the message is DFHWB0127. For any other program, the message is DFHWB0128. In any case an exception trace entry (trace point A116) is written.
- The EXEC CICS LINK to the program specified in **wba1_server_program_name** received an INVREQ or a LENGERR or an unexpected response. The business logic interface writes an exception trace entry (trace point A115) and issues a message (DFHWB0120).

501 One of the following occurred:

- Decode returned a response of URP_EXCEPTION with an undefined reason code. The business logic interface writes an exception trace entry (trace point A11A) and issues a message (DFHWB0121).
- Decode returned a response of URP_INVALID. The business logic interface writes an exception trace entry (trace point A11B) and issues a message (DFHWB0121).
- Decode returned a response of URP_DISASTER. The business logic interface writes an exception trace entry (trace point A11C) and issues a message (DFHWB0121).
- Decode returned an undefined response. The business logic interface writes an exception trace entry (trace point A11D) and issues a message (DFHWB0121).
- Encode returned a response of URP_EXCEPTION with an undefined reason code. The business logic interface writes an exception trace entry (trace point A11A) and issues a message (DFHWB0122).
- Encode returned a response of URP_INVALID. The business logic interface writes an exception trace entry (trace point A11B) and issues a message (DFHWB0122).

- Encode returned a response of URP_DISASTER. The business logic interface writes an exception trace entry (trace point A11C) and issues a message (DFHWB0122).
- Encode returned an undefined response. The business logic interface writes an exception trace entry (trace point A11D) and issues a message (DFHWB0122).

503 One of the following occurred:

- The EXEC CICS LINK to the program specified in **wba1_server_program_name** received a TERMERR response. The business logic interface writes an exception trace entry (trace point A114) and issues a message (DFHWB0120).
- The EXEC CICS LINK to the program specified in **wba1_server_program_name** received a SYSIDERR or ROLLEDBACK response. The business logic interface writes an exception trace entry (trace point A115) and issues a message (DFHWB0120).

Appendix B. Messages and Codes

DFHWB0001 *applid* **An abend (code** *aaa/bbbb***) has occurred at offset** *X'offset'* **in module** *modname***.**

Explanation: An abnormal end (abend) or program check has occurred in module *modname*. This implies that there may be an error in the CICS code. Alternatively, unexpected data has been input, or storage has been overwritten.

The code *aaa/bbbb* is a 3-digit hexadecimal MVS code (if applicable), followed by a 4-digit alphanumeric CICS code. The MVS code is a system completion code (for example, 0C1 or D37). If an MVS code is not applicable, this field is filled with three hyphens. The CICS code is an abend code or a number referring to a CICS message (for example, AKEA is a CICS abend code; 1310 refers to message DFHTS1310).

System Action: An exception entry is made in the trace table. A system dump is taken, unless you have specifically suppressed dumps in the dump table.

CICS continues unless you have specified in the dump table that CICS should terminate. If appropriate, an error return code is sent to the caller of this domain. In this case CICS could be terminated by the caller (for example, the domain manager, DFHDMDM). A message is issued to this effect.

Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: Notify the system programmer. If CICS is still running, it is necessary to decide whether to terminate CICS.

Look up the MVS code, if there is one, in the relevant MVS codes manual.

Next, look up the CICS alphanumeric code. This tells you, for example, whether the error was a program check, an abend, or a runaway, and may give you some guidance concerning user response.

If module *modname* is not crucial to the running of your CICS system, you may decide to continue to run and bring CICS down at a convenient time to resolve the problem. If you cannot run without the full use of module *modname* you should bring CICS down in a controlled shutdown.

You need further assistance from IBM to resolve this problem. See part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed.

Destination: Console

Module: DFHWBWB

XMEOUT Parameters: *applid, aaa/bbbb, X'offset', modname*

DFHWB0002 *applid* **A severe error (code** *X'code'***) has occurred in module** *modname***.**

Explanation: An error has been detected in module *modname*. The code *X'code'* is the exception trace point ID, which gives an indication of the cause of the error:

- 9F08—Web Interface Global Work Area could not be found
- 9F09—Error initializing C environment
- 9F0A—Insufficient storage to process incoming request
- 9F0B—Error on **socket** call to TCP/IP for MVS
- 9F0C—Error on **gethostid** call to TCP/IP for MVS
- 9F0D—Requested port already in use
- 9F0E—Requested port was not available
- 9F0F—Error on **bind** call to TCP/IP for MVS
- 9F10—Error on **ioctl** call to TCP/IP for MVS
- 9F11—Logic error occurred
- 9F12—Error on **listen** call to TCP/IP for MVS
- 9F13—Error deleting CICS lock
- 9F14—Duplicate CICS lock requested
- 9F15—Error acquiring lock
- 9F16—Bad parameter list passed to C component
- 9F19—Client for HTTP request could not be identified
- 9F1A—Error on **send** call to TCP/IP for MVS
- 9F1B—Error on **getsockopt** call to TCP/IP for MVS
- 9F1C—Error on **close** call to TCP/IP for MVS
- 9F1D—Error on **select** call to TCP/IP for MVS
- 9F1E—Error on **accept** call to TCP/IP for MVS
- 9F20—Invalid socket on **receive** call to TCP/IP for MVS
- 9F21—

APAR PQ05765
Removed the words ″during receive″, RGW 4 Nov 1997

TCP/IP connection lost

- 9F22—Insufficient storage available to perform TCP/IP for MVS **receive**
- 9F24—Invalid socket on **send** call to TCP/IP for MVS
- 9F25—Insufficient storage available to perform TCP/IP for MVS **send**
- 9F27—Error on **gethostname** call to TCP/IP for MVS
-

APAR PQ08695
Added error codes 9F28 to 9F2D — RGW 22/06/1998

9F28 — Error on **receive** call to TCP/IP for MVS
- 9F29 — No data returned from **receive** call to TCP/IP for MVS
- 9F2A — Non-HTTP request received
- 9F2B — Receive buffer too small for data received
- 9F2C — Error on **select** call to TCP/IP for MVS
- 9F2D — Error on **gethostbyaddr** call to TCP/IP for MVS
- 9F33—

APAR PQ21555
Added 9F33 MJL 4/99

Runaway detected on a **send** call to TCP/IP for MVS
- 9F35—

APAR PQ24024
Added 9F35 MJL 6/99

Error on **select** call to TCP/IP for MVS

System Action: An exception entry (code *X'code'* in the message) is made in the trace table. A system dump is taken, unless you have specifically suppressed dumps in the dump table.

Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response:

There are some specific user actions that can be taken for certain values of *code*:

- 9F0D,9F0E: If these errors occur, you may be trying to enable the Web Interface using a port which TCP/IP for MVS has already reserved for another service. Look at the TCP/IP for MVS diagnostics.
- 9F22,9F25: If these short on storage conditions persist, you may need to adjust the CICS DSA size, or maximum number of tasks, or adjust your TCP/IP for MVS.
- 9F27: Check that TCP/IP for MVS has been started.

For other TCP/IP for MVS problems, look at the TCP/IP for MVS diagnostics.

You may need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed.

Destination: Console

Module: DFHWBWB

XMEOUT Parameters: *applid, X'code', modname*

DFHWB0100 *date time applid tranid* **The CICS Web Interface program cannot link to program DFHWBA1. EIBRESP:** *eibresp* **EIBRESP2:** *resp2val* **Host IP address:** *hostaddr***. Client IP address:** *clientaddr***.**

Explanation: The alias program used EXEC CICS LINK but was unable to link to program DFHWBA1.

System Action: The link is abandoned. An HTTP response code of 500 (internal server error) is returned to the Web Browser. The alias abends with abend code AWBL.

User Response: Use the CEDA transaction to ensure that program DFHWBA1 has been correctly defined and installed.

Destination: CWBO

Module: DFHWBA

XMEOUT Parameters: *date, time, applid, tranid, eibresp, resp2val, hostaddr, clientaddr*

DFHWB0101 *date time applid tranid* **The CICS Web Interface alias program DFHWBA detected a failure in program DFHWBA1. Host IP address:** *hostaddr***. Client IP address:** *clientaddr***.**

Explanation: Program DFHWBA1 has returned an error response to the alias.

System Action: The request is abandoned. The error response returned by program DFHWBA1 is returned to the Web Browser in an HTTP response:

403 The userid associated with the request is not authorized to invoke the requested converter program, or the requested server program.

404 A link to the converter program or to the server program failed because CICS could not locate the requested program.

500 A link to the converter program or to the server program failed with an unexpected error.

503 A link to the converter program or to the server program failed for one of the following reasons:
- The server program is defined as remote, but the link to this program failed with a SYSID error, so the remote connection is either not defined correctly, or not active.
- The link to the converter or the server program failed with the ROLLEDBACK response.

The alias abends with abend code AWBM.

User Response: Check program DFHWBA1 and the programs which it calls.

Destination: CWBO

Module: DFHWBA

XMEOUT Parameters: *date, time, applid, tranid, hostaddr, clientaddr*

DFHWB0102 *date time applid tranid* **The CICS Web Interface alias program has received an incorrect response on a call made to CICS during alias initialization. EIBRESP:** *eibresp* **EIBRESP2:** *resp2val***.**

Explanation: The alias program has received an unexpected response on a call made to CICS during alias initialization.

System Action: The alias abends with abend code AWBI.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed.

Destination: CWBO

Module: DFHWBA

XMEOUT Parameters: *date, time, applid, tranid, eibresp, resp2val*

DFHWB0103 *date time applid tranid* **The CICS Web Interface alias program has received an incorrect response on a call made to CICS during alias initialization. EIBRESP:** *eibresp* **EIBRESP2:** *resp2val***.**

Explanation: The alias program detected an error response on RETRIEVE for the START data.

System Action: The alias abends with abend code AWBF.

User Response: The alias program DFHWBA is only to be used for alias transactions started by the CICS Web Interface. User-written applications should not be starting alias transactions, as data passed to the alias will not be in the expected format.

If CICS is experiencing problems with temporary storage, this may have caused the start data for the alias to be lost. See any associated CICS messages to help with problem diagnosis.

Destination: CWBO

Module: DFHWBA

XMEOUT Parameters: *date, time, applid, tranid, eibresp, resp2val*

DFHWB0104 *date time applid tranid* **The CICS Web Interface alias program has been unable to continue processing this client request.**

Explanation: The alias program has detected that the CICS Web Interface may have been disabled since this client request was scheduled by the server controller. This is indicated by an incorrect reference to the GWA.

System Action: The client request is abandoned, and no reply is sent to the client. The alias abends with abend code AWBG.

User Response: Check that the CICS Web Interface has not been disabled since this client request was first scheduled. This problem may arise when long-running CICS programs are being used. It may also occur if the interface is disabled and immediately re-enabled.

Destination: CWBO

Module: DFHWBAS

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB0105 *date time applid tranid* **The CICS Web Interface has encountered a severe internal error while processing this client request. Host IP address:** *hostaddr***. Client IP address:** *clientaddr***.**

Explanation: The alias is unable to switch TCB modes to allow it to send a reply to the client. The RP TCB is not active.

System Action: The client request is abandoned, and no reply is sent to the client. A system dump is taken. The alias abends with abend code AWBJ. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: See any associated CICS messages to help with problem diagnosis. If you cannot determine why the TCB mode could not be switched, you need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBAS

XMEOUT Parameters: *date, time, applid, tranid, hostaddr, clientaddr*

DFHWB0106 *date time applid tranid* **The CICS Web Interface program DFHWBA has detected an error.**

Explanation: The alias had detected an error.

System Action: A system dump is taken. The alias

abends with abend code AWBH. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: Examine the diagnostics to determine the reason for the error.

Destination: CWBO

Module: DFHWBAS

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB0107 *date time applid tranid* **The CICS Web Interface alias program has detected a FREEMAIN error. EIBRESP:** *eibresp* **EIBRESP2:** *resp2val* **Host IP address:** *hostaddr***. Client IP address:** *clientaddr***.**

Explanation: The CICS alias has detected a FREEMAIN error and an EIB response was returned.

System Action: Processing continues.

User Response: None

Destination: CWBO

Module: DFHWBAS

XMEOUT Parameters: *date, time, applid, tranid, eibresp, resp2val, hostaddr, clientaddr*

DFHWB0108 *date time applid tranid* **The CICS Web Interface alias program has detected an abend. Host IP address:** *hostaddr* **Client IP address:** *clientaddr***.**

Explanation: The alias has detected an abend.

System Action: The alias abends with abend code AWBK.

User Response: Examine the diagnostics to determine the reason for the error.

Destination: CWBO

Module: DFHWBAS

XMEOUT Parameters: *date, time, applid, tranid, hostaddr, clientaddr*

DFHWB0120 *date time applid tranid* **The CICS Web Interface program DFHWBA1 cannot link to program** *program_name***. EIBRESP:** *eibresp* **EIBRESP2:** *resp2val***.**

Explanation: Program DFHWBA1 used an EXEC CICS LINK but was unable to link to the given program and an EIB response was returned.

System Action: The link is abandoned.

User Response: Ensure that the program definition is correct.

Destination: CWBO

Module: DFHWBA1

XMEOUT Parameters: *date, time, applid, tranid, program_name, eibresp, resp2val*

DFHWB0121 *date time applid tranid* **The CICS Web Interface program DFHWBA1 encountered an error during Decode processing in the converter** *program***. Error code:** *X'errorid'***.**

Explanation: The Decode function of the converter has returned an error.

System Action: An error message is sent to the client.

User Response: Examine the diagnostics to determine the reason for the error. The error code issued in the message is equivalent to the trace point ID issued with this error.

Destination: CWBO

Module: DFHWBA1

XMEOUT Parameters: *date, time, applid, tranid, program, X'errorid'*

DFHWB0122 *date time applid tranid* **The CICS Web Interface program DFHWBA1 encountered an error during Encode processing in the converter** *program***. Error code:** *X'errorid'***.**

Explanation: The Encode function of the converter program has returned an error.

System Action: An error message is sent to the client. The error code issued in the message is equivalent to the trace point ID issued with this error.

User Response: Examine the diagnostics to determine the reason for the error.

Destination: CWBO

Module: DFHWBA1

XMEOUT Parameters: *date, time, applid, tranid, program, X'errorid'*

DFHWB0123 *date time applid tranid* **The CICS Web Interface program DFHWBA1 has detected an error.**

Explanation: Program DFHWBA1 has detected an error.

System Action: A system dump is taken. The transaction abends with abend code AWBR. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: Examine the diagnostics to determine the reason for the error.

Destination: CWBO

Module: DFHWBA1

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB0124 *date time applid tranid* **The CICS Web Interface program DFHWBA1 has been started incorrectly.**

Explanation: Program DFHWBA1 has detected an error while validating initialization information. This probably means that the program has been started incorrectly.

System Action: The transaction abends with abend code AWBQ.

User Response: Check that the program was not started by a transient data trigger level or by a CECI user.

Destination: CWBO

Module: DFHWBAS

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB0125 *date time applid tranid* **The CICS Web Interface program DFHWBA1 has detected an abend issued by the program** *program***.**

Explanation: Program DFHWBA1 has detected an abend by the program that was servicing the request.

System Action: The alias returns control to the caller.

User Response: Examine the diagnostics to determine the reason for the error.

Destination: CWBO

Module: DFHWBA1S

XMEOUT Parameters: *date, time, applid, tranid, program*

DFHWB0126 *date time applid tranid* **The CICS Web Interface program DFHWBA1 has detected an abend issued by Encode in converter program** *program***.**

Explanation: Program DFHWBA1 has detected an abend by the program that was servicing the request during Encode processing.

System Action: The alias returns control to the caller.

User Response: Examine the diagnostics to determine the reason for the error.

Destination: CWBO

Module: DFHWBA1S

XMEOUT Parameters: *date, time, applid, tranid, program*

DFHWB0127 *date time applid tranid* **The CICS Web Interface program DFHWBA1 has detected an abend issued by Decode in converter** *program***.**

Explanation: Program DFHWBA1 has detected an abend by the converter that was servicing the request during Decode processing.

System Action: The alias returns control to the caller.

User Response: Examine the diagnostics to determine the reason for the error.

Destination: CWBO

Module: DFHWBA1S

XMEOUT Parameters: *date, time, applid, tranid, program*

DFHWB0128 *date time applid tranid* **An error has been detected by program** *program***.**

Explanation: Program DFHWBA1 has detected an error.

System Action: The alias returns control to the caller.

User Response: Examine the diagnostics to determine the reason for the error.

Destination: CWBO

Module: DFHWBA1S

XMEOUT Parameters: *date, time, applid, tranid, program*

DFHWB0500I
date time applid tranid **CICS Web Interface enable processing is complete. Host IP address:** *hostaddr***.**

Explanation: The enable process has completed successfully.

System Action: Processing continues.

User Response: None.

Destination: Console and Transient Data Queue CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, hostaddr*

DFHWB0501 *date time applid tranid* **CICS Web Interface normal disable processing has started. Host IP address:** *hostaddr***.**

Explanation: The server controller has started normal disable processing following a request by a connection manager user.

System Action: Processing continues.

User Response: None.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, hostaddr*

DFHWB0502 *date time applid tranid* **CICS Web Interface immediate disable processing has started. Host IP address:** *hostaddr***.**

Explanation: The server controller has started immediate disable processing following a request by a connection manager user.

System Action: Processing continues.

User Response: None.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, hostaddr*

DFHWB0503I
date time applid tranid **CICS Web Interface disable processing is complete.**

Explanation: The server controller has completed the disable processing.

System Action: Processing continues.

User Response: None.

Destination: Console and Transient Data Queue CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB0504 *date time applid tranid* **The CICS Web Interface server controller has started exception disable of the CICS Web Interface. Host IP address:** *hostaddr***.**

Explanation: The server controller has started an exception disable of the CICS Web Interface following an error during its operation. The error has already been reported.

System Action: Disable processing continues.

User Response: See the associated diagnostics for further information about the error.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, hostaddr*

DFHWB0510 *date time applid tranid* **The CICS Web Interface server controller is abending with abend code AWB1.**

Explanation: The server controller encountered an error and cannot continue.

System Action: The server controller abends with abend code AWB1. The CICS Web Interface is disabled.

User Response: This message will always be preceded by an error message that describes the particular error encountered. See the associated diagnostics and the description of that message for further guidance.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB0511 *date time applid tranid* **The CICS Web Interface server controller has received an unexpected response on a call to CICS during enable processing. EIBRESP:** *eibresp* **EIBRESP2:** *resp2val***.**

Explanation: Enable processing cannot continue because of an error in an EXEC CICS EXTRACT EXIT call. The response values returned by the call are included in the message.

System Action: A system dump is taken. The server controller abends with abend code AWB1. The CICS Web Interface remains disabled. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, eibresp, resp2val*

DFHWB0512 *date time applid tranid* **The CICS Web Interface could not be enabled because of an internal error in the server controller.**

Explanation: The server controller cannot start because of an internal error.

System Action: A system dump is taken. The server controller abends with abend code AWB1. The CICS Web Interface remains disabled. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB0513 *date time applid tranid* **The CICS Web Interface server controller is abending with abend code AWB2.**

Explanation: The server controller encountered an error and initiated an exception disable.

System Action: The server controller abends with abend code AWB2. The CICS Web Interface is disabled.

User Response: This message will always be preceded by an error message that describes the particular error encountered. See the associated diagnostics and the description of that message for further guidance.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB0520 *date time applid tranid* **The CICS Web Interface server controller received a program ID error when linking to analyzer** *progname***. EIBRESP:** *eibresp***. EIBRESP2:** *resp2val***. Host IP address:** *hostaddr***. Client IP address:** *clientaddr***.**

Explanation: The server controller used EXEC CICS LINK for analyzer but received a PGMIDERR response.

System Action: An error response is sent to the client and processing of the request is terminated.

User Response: Make sure that the analyzer name set by connection manager at enable time is valid and is a currently installed program.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, progname, eibresp, resp2val, hostaddr, clientaddr*

DFHWB0521 *date time applid tranid* **The CICS Web Interface server controller received a NOTAUTH condition when linking to analyzer** *progname***. EIBRESP:** *eibresp***. EIBRESP2:** *resp2val***. Host IP address:** *hostaddr***. Client IP address:** *clientaddr***.**

Explanation: The server controller used EXEC CICS LINK to link to analyzer but received a NOTAUTH response.

System Action: An error response is sent to the client and processing of the request is terminated.

User Response: Make sure that the server controller task is authorized to use the specified analyzer.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, progname, eibresp, resp2val, hostaddr, clientaddr*

DFHWB0522 *date time applid tranid* **The CICS Web Interface server controller received a serious error when linking to analyzer** *progname***. EIBRESP:** *eibresp***. EIBRESP2:** *resp2val***. Host IP address:** *hostaddr***. Client IP address:** *clientaddr***.**

Explanation: The server controller used EXEC CICS LINK to link to analyzer but received an error response other than PGMIDERR or NOTAUTH.

System Action: An error response is sent to the client and processing of the request is terminated.

User Response: Analyze the EIBRESP and EIBRESP2 values in the message to determine the cause of the link failure.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, progname, eibresp, resp2val, hostaddr, clientaddr*

DFHWB0523 *date time applid tranid* **The CICS Web Interface analyzer program returned an error response. Program name:** *progname***. RESPONSE:** *response***. REASON:** *reason***. Host IP address:** *hostaddr***. Client IP address:** *clientaddr***.**

Explanation: As part of its normal processing of a request, the server controller invokes the user replaceable analyzer to tailor the required actions. This program returns RESPONSE and REASON values. The values returned for this request indicate that an error has been detected by the analyzer.

System Action: An error response is sent to the client and processing of the request is terminated.

User Response: Examine the RESPONSE and REASON values in the message to determine the cause of the error.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, progname, response, reason, hostaddr, clientaddr*

DFHWB0524 *date time applid tranid* **The CICS Web Interface server controller received a NOTAUTH condition when starting the alias transaction** *tranid***. EIBRESP:** *eibresp***. EIBRESP2:** *resp2val***. Termid:** *termid***. Userid:** *userid***. Host IP address:** *hostaddr***. Client IP address:** *clientaddr***.**

Explanation: The server controller issued EXEC CICS START for the requested alias transaction ID *tranid* but received a NOTAUTH response.

System Action: An error response is sent to the client and processing of the request is terminated.

User Response: Ensure that the server controller task is authorized to use the specified transaction ID.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, tranid, eibresp, resp2val, termid, userid, hostaddr, clientaddr*

DFHWB0525 *date time applid tranid* **The CICS Web Interface server controller received a TERMIDERR condition when starting the alias transaction** *tranid***. EIBRESP:** *eibresp***. EIBRESP2:** *resp2val***. Termid:** *termid***. Userid:** *userid***. Host IP address:** *hostaddr***. Client IP address:** *clientaddr***.**

Explanation: The server controller issued EXEC CICS START for the requested alias transaction ID *tranid* but received a TERMIDERR response.

System Action: An error response is sent to the client and processing of the request is terminated.

User Response: Ensure that the specified terminal has an installed definition, or change the analyzer to request either a different terminal or none at all.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, tranid, eibresp, resp2val, termid, userid, hostaddr, clientaddr*

DFHWB0526 *date time applid tranid* **The CICS Web Interface server controller received a TRANSIDERR condition when starting the alias transaction** *tranid***. EIBRESP:** *eibresp***. EIBRESP2:** *resp2val***. Termid:** *termid***. Userid:** *userid***. Host IP address:** *hostaddr***. Client IP address:** *clientaddr***.**

Explanation: The server controller issued EXEC CICS START for the requested alias transaction ID *tranid* but received a TRANSIDERR response.

System Action: An error response is sent to the client and processing of the request is terminated.

User Response: Ensure that the specified transaction has an installed definition, or change the analyzer to request a different transaction ID.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, tranid, eibresp, resp2val, termid, userid, hostaddr, clientaddr*

DFHWB0527 *date time applid tranid* **The CICS Web Interface server controller received a USERIDERR condition when starting the alias transaction** *tranid***. EIBRESP:** *eibresp***. EIBRESP2:** *resp2val***. Termid:** *termid***. Userid:** *userid***. Host IP address:** *hostaddr***. Client IP address:** *clientaddr***.**

Explanation: The server controller issued EXEC CICS START for the requested alias transaction ID *tranid* but received a USERIDERR response.

System Action: An error response is sent to the client and processing of the request is terminated.

User Response: Ensure that the specified user ID is valid, or change the analyzer to request either a different user ID or none at all.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, tranid, eibresp, resp2val, termid, userid, hostaddr, clientaddr*

DFHWB0528 *date time applid tranid* **The CICS Web Interface server controller received a serious error condition when starting the alias transaction** *tranid***. EIBRESP:** *eibresp***. EIBRESP2:** *resp2val***. Termid:** *termid***. Userid:** *userid***. Host IP address:** *hostaddr***. Client IP address:** *clientaddr***.**

Explanation: The server controller issued EXEC CICS START for the requested alias transaction ID *tranid* but received an error response that was not caused by invalid parameters returned from the analyzer.

System Action: An error response is sent to the client and processing of the request is terminated.

User Response: Use the EIBRESP and EIBRESP2 values to determine why the EXEC CICS START failed.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, tranid, eibresp, resp2val, termid, userid, hostaddr, clientaddr*

DFHWB0529 *date time applid tranid* **The CICS Web Interface server controller HANDLE ABEND code was entered as a result of an error in the analyzer** *analyzer_program_name***. Client IP address:** *clientaddr* **Host IP address:** *hostaddr***.**

Explanation: An error has occurred in the analyzer. Because the analyzer does not contain HANDLE ABEND logic, the error is routed to the server controller.

System Action: An error is returned to the client and the request is terminated.

User Response: Correct the error in the analyzer. Add handle abend logic to the analyzer so that it can handle its own errors, then replace it.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, analyzer_program_name, clientaddr, hostaddr*

DFHWB0530 *date time applid tranid* **The CICS Web Interface server controller has encountered an internal error while processing a client request. Client IP address:** *clientaddr* **Host IP address:** *hostaddr***.**

Explanation: An internal error has forced the CICS Web Interface to abandon a client request.

System Action: A system dump is taken. An error response is sent to the client and the request is terminated. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, clientaddr, hostaddr*

DFHWB0531 *date time applid tranid* **The CICS Web Interface server controller has encountered an internal error when no client request was being processed. Host IP address:** *hostaddr***.**

Explanation: An internal error has occurred in the server controller. No client requests are affected.

System Action: A system dump is taken. The server controller continues. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, hostaddr*

DFHWB0532 *date time applid tranid* **The CICS Web Interface server controller has received an unexpected response on a call to CICS during disable processing. EIBRESP:** *eibresp* **EIBRESP2:** *resp2val***.**

Explanation: Disable processing received an error in an EXEC CICS DISABLE call. The response values returned by the call are included in the message.

System Action: Disable continues, but will not be complete. Any attempt to re-enable the CICS Web Interface will probably fail. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, eibresp, resp2val*

DFHWB0533 *date time applid tranid* **The CICS Web Interface server controller has received an unexpected response on a call to CICS during disable processing. EIBRESP:** *eibresp* **EIBRESP2:** *resp2val***.**

Explanation: Disable processing received an error in an EXEC CICS FREEMAIN call to remove the HTTP caller module, DFHWBWB, from storage. The response values returned by the call are included in the message.

System Action: Disable continues, but will not be complete. Any attempt to re-enable the CICS Web Interface will probably fail. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for*

MVS/ESA Problem Determination Guide for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, eibresp, resp2val*

DFHWB0534 *date time applid tranid* **The CICS Web Interface server controller could not activate the RP TCB during enable processing.**

Explanation: Enable processing received an error while trying to activate the RP TCB.

System Action: The server controller abends with abend code AWB1. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: See the messages already issued by the dispatcher domain to determine the cause of the error.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB0535 *date time applid tranid* **The CICS Web Interface server controller could not add a storage subpool during enable processing.**

Explanation: Enable processing received an error from a storage manager ADD_SUBPOOL request.

System Action: The server controller abends with abend code AWB1. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: See the messages already issued by the storage manager domain to determine the cause of the error.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB0536 *date time applid tranid* **The CICS Web Interface server controller could not load module DFHWBWB during enable processing. EIBRESP:** *eibresp* **EIBRESP2:** *resp2val***.**

Explanation: Enable processing received an error from an EXEC CICS LOAD call for the HTTP caller program, DFHWBWB. The response values returned by the call are included in the message.

System Action: The server controller abends with abend code AWB1.

User Response: Ensure that DFHWBWB is defined to CICS and is present in the DFHRPL library.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, eibresp, resp2val*

DFHWB0537 *date time applid tranid* **The CICS Web Interface server controller could not initialize the HTTP caller during enable processing.**

Explanation: Enable processing received an error while trying to initialize the HTTP caller component.

System Action: The server controller abends with abend code AWB1.

User Response: The HTTP caller issues messages indicating the cause of the error.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB0538 *date time applid tranid* **The CICS Web Interface server controller has encountered an internal error when no client request was being processed. Host IP address:** *hostaddr***.**

Explanation: The HTTP caller detected an error while processing a call from the server controller.

System Action: The HTTP caller issues diagnostic messages. An exception disable of the CICS Web Interface is initiated. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: Proceed as instructed in the HTTP caller messages.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, hostaddr*

DFHWB0539 *date time applid tranid* **The CICS Web Interface server controller has encountered an internal error when no client request was being processed. Host IP address:** *hostaddr***.**

Explanation: the CICS dispatcher detected an error while processing a call from the server controller.

System Action: The dispatcher issues diagnostic messages. An exception disable of the CICS Web

Interface is initiated. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: Proceed as indicated in the dispatcher messages.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, hostaddr*

DFHWB0540 *date time applid tranid* **The CICS Web Interface server controller has encountered an internal error.**

Explanation: The server controller was unable to switch to the RP TCB.

System Action: The CICS dispatcher issues messages indicating the cause of the error. An exception disable of the CICS Web Interface is initiated.

User Response: Proceed as indicated in the dispatcher messages.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB0541 *date time applid tranid* **The CICS Web Interface server controller has encountered an internal error.**

Explanation: The server controller was unable to acquire the storage to be used for the next client request.

System Action: The storage manager issues messages indicating the cause of the error. An exception disable of the CICS Web Interface is initiated.

User Response: Proceed as indicated by the storage manager messages.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB0550 *date time applid tranid* **The CICS Web Interface server controller could not initialize the HTTP caller during enable processing because the requested port was not available.**

Explanation: The CICS Web Interface could not be enabled, because the requested port was not available.

System Action:

APAR PQ06710
Changed from AWBA to AWB1 by RGW 4 Nov 1997

The server controller abends with abend code AWB1.

User Response: Check your TCP/IP for MVS/ESA system. The requested port may already be in use, or may have been reserved for use by another server.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1000 *date time applid* **The CICS Web Interface HTTP caller is initializing.**

Explanation: The server controller has started initialization of the HTTP caller.

System Action: Processing continues.

User Response: None.

Destination: CWBO

Module: DFHWBWB

XMEOUT Parameters: *date, time, applid*

DFHWB1001 *date time applid* **The CICS Web Interface HTTP caller has been initialized successfully.**

Explanation: The HTTP caller has been initialized, and it is now ready to process incoming HTTP requests.

System Action: Processing continues.

User Response: None.

Destination: CWBO

Module: DFHWBWB

XMEOUT Parameters: *date, time, applid*

DFHWB1002 *date time applid* **The CICS Web Interface HTTP caller is shutting down.**

Explanation: Termination of the HTTP caller has started.

System Action: Processing continues.

User Response: None.

Destination: CWBO

Module: DFHWBWB

XMEOUT Parameters: *date, time, applid*

DFHWB1003 *date time applid* **The CICS Web Interface HTTP caller has successfully shut down.**

Explanation: Termination of the HTTP caller has completed.

System Action: Processing continues.

User Response: None.

Destination: CWBO

Module: DFHWBWB

XMEOUT Parameters: *date, time, applid*

DFHWB1004 *date time applid* **CICS has detected that TCP/IP is no longer active. The CICS Web Interface will be disabled.**

Explanation: TCP/IP is no longer active. Requests cannot be processed by the CICS Web Interface.

System Action: Exception entry code X'9F36' is made in the trace table. No system dump is taken, unless you have specifically requested dumps in the dump table. The CICS Web Interface is exception disabled.

User Response: The CICS Web Interface should not be re-enabled using the CWBC transaction until TCP/IP has been restarted.

Destination: CWBO

Module: DFHWBWB

XMEOUT Parameters: *date, time, applid*

DFHWB1005 *date time applid tranid* **An error (code** *X'code'***) has occurred in module** *modname* **whilst processing data received from IP address** *ipaddr***.**

Explanation: An error has been detected in module *modname* while processing data received from IP address *ipaddr*. The code *X'code'* is the exception trace point ID, which gives an indication of the cause of the error:

- 9F21—

APAR PQ05765
Added 9F21, RGW 4 Nov 1997

TCP/IP connection lost

- 9F22—Storage error detected on receive call to TCP/IP for MVS
- 9F24—Invalid socket detected on send call to TCP/IP for MVS
- 9F25—Storage error detected on send call to TCP/IP for MVS
- 9F29—No data received on receive call to TCP/IP for MVS
- 9F2B—Amount of data received on receive to TCP/IP for MVS exceeded the limit for the CICS Web Interface.

System Action: Exception entry code *X'code'* is made in the trace table. No system dump is taken, unless you have specifically requested dumps in the dump table.

User Response: There are some specific user actions for certain values of *X'code'*:

- 9F0D,9F0E: If these errors occur, you may be trying to enable the Web Interface using a port which TCP/IP for MVS has already reserved for another service. Look at the TCP/IP for MVS diagnostics.
- 9F22,9F25: If these short on storage conditions persist, you may need to adjust the CICS DSA size or maximum number of tasks, or adjust your TCP/IP for MVS.
- 9F27: Ensure that TCP/IP for MVS has been started.

For other TCP/IP for MVS problems, look at the TCP/IP for MVS diagnostics.

Destination: CWBO

Module: DFHWBWB

XMEOUT Parameters: *date, time, applid, tranid, X'code', modname, ipaddr*

DFHWB1006 *date time applid tranid* **The CICS Web Interface is unable to resolve the TCP/IP name of the machine on which it is running. Error Code** *X'code'* **IP Address:***ipaddr***.**

Explanation:

APAR PQ11796
New message

An error has been detected in module *modname* following a TCP/IP GETHOSTBYADDR call to retrieve the IP name of the host (*ipaddr*) on which CICS is running. The code is the error code returned by TCP/IP, which gives an indication of the cause of the error. Refer to the *REQTEXT* for details.

System Action: Exception entry code 9F2D is made in the trace table. No system dump is taken, unless you have specifically requested dumps in the dump table.

User Response: This error occurs if there is no TCP/IP Domain Name Server set up for the instance of TCP/IP for MVS which CICS is using. Users who are not using Domain Name Server need take no action. Users who are using a Domain Name Server, and who wish the CICS Web Interface to have access to the full name of the instance of TCP/IP for MVS that the CICS Web Interface is using, need to use the code to determine the reason for the failure of the

GETHOSTBYADDR call. Refer to the *REQTEXT* for details.

Destination: CWBO

Module: DFHWBWB

XMEOUT Parameters: *date, time, applid, tranid, X'code', ipaddr*

DFHWB1100 E
date time applid **The CICS Web Interface received data from the user application that is longer than expected.**

Explanation: The environment variables program has received data from a user application. However the data received was longer than expected.

System Action: Exception trace point APA302 is written. The environments program abnormally terminates with abend code AWE1.

User Response: Examine the data sent to CICS from the application program.

Destination: Console Routecodes 2 and 12 and Transient Data Queue CWBO

Module: DFHWBENV

XMEOUT Parameters: *date, time, applid*

DFHWB1200 *date time applid tranid* **The CICS Web Interface analyzer program set parameter wbra_user_data_length to more than the maximum. Program name: *progname*. RESPONSE: *response*. REASON: *reason*. Host IP address: *hostaddr*. Client IP address: *clientaddr*. Data offset:*data offset*. Data length: *data length*. Buffer length: *buffer length*.**

Explanation: As part of its normal processing of a request, the server controller invokes the user replaceable analyzer to tailor the required actions. This program is passed the length of the user data part of the HTTP request in parameter **wbra_user_data_length**, which it can modify. However, the modified value is greater than the maximum allowable value which represents the available space in the data buffer.

System Action: An error response is sent to the client and processing of the request is terminated.

User Response: Modify the analyzer program so that it does not set the parameter wbra_user_data_length to be greater than the maximum. The sum of the data offset and the data length should not exceed the buffer length.

Destination: CWBO

Module: DFHWBM

XMEOUT Parameters: *date, time, applid, tranid, progname, response, reason, hostaddr, clientaddr, data offset, data length, buffer length*

DFHWB1500 *date time applid tranid* **Invalid data has been entered in field** *fieldname*.

Explanation: Invalid data was entered on a connection manager panel in field *fieldname.*

System Action: The panel is redisplayed and the field in error is highlighted.

User Response: Enter valid data in the field indicated. See "Chapter 5. Configuration with the connection manager" on page 55 for further guidance.

Destination: Terminal End User

Module: DFHWBC0B

DFHWB1505 *date time applid tranid* **The CICS Web Interface connection manager has not been started correctly.**

Explanation: The connection manager has been started from a non-BMS terminal but is not being used to enable or disable the CICS Web Interface.

System Action: The connection manager terminates.

User Response: See "Chapter 5. Configuration with the connection manager" on page 55 for guidance on how to start the connection manager.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1506 *date time applid tranid* **The CICS Web Interface connection manager detected an error attempting to retrieve fast path data. EIBRESP: *eibresp*.**

Explanation: The connection manager was attempting to retrieve any fast path commands that may have been specified when it was initiated. The connection manager issued an EXEC CICS GETMAIN command, but received the response *eibresp.*

System Action: A system dump is taken. The connection manager continues but any fast path commands are ignored. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid, eibresp*

DFHWB1507 *date time applid tranid* **An invalid CICS Web Interface fast path command has been entered:** *fastpath_command***.**

Explanation: The connection manager was started by entering a fast path command, but the format of the command was invalid.

System Action: The connection manager is started, but fast path commands are ignored.

User Response: Enter a valid fast path command. See "Chapter 5. Configuration with the connection manager" on page 55 for further guidance.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid, fastpath_command*

DFHWB1508 *date time applid tranid* **The CICS Web Interface connection manager has not been started correctly.**

Explanation: The connection manager was attempting to retrieve any fast path commands that may have been specified when it was initiated, but detected an invalid STARTCODE indicator.

System Action: The connection manager continues but any fast path commands are ignored.

User Response: See "Chapter 5. Configuration with the connection manager" on page 55 for guidance on how to start the connection manager. If the connection manager was started correctly, you need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1509 *date time applid tranid* **The CICS Web Interface connection manager detected an error attempting to retrieve any fast path data. EIBRESP:** *eibresp***.**

Explanation: The connection manager was attempting to retrieve any fast path commands that may have been specified when it was initiated using an EXEC CICS START command. The connection manager issued an EXEC CICS RETRIEVE command, but received the response *eibresp*.

System Action: A system dump is taken. The connection manager continues but any fast path commands are ignored. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid, eibresp*

DFHWB1510 *date time applid tranid* **The CICS Web Interface connection manager detected an error while accessing the CICS Web Interface data set, CICS file** *filename***. EIBRESP:** *eibresp***.**

Explanation: The connection manager could not access the CICS Web Interface data set, CICS file *filename*. An EXEC CICS READ was issued, but received the response *eibresp*. The data set has not been correctly defined to CICS for one of the following reasons:

- No file definition has been found for *filename*. The CICS Web Interface has therefore not been installed correctly.
- READ operations are not allowed on the file.
- The file is DISABLED, either due to an incorrect file definition, or due to operator intervention.
- The file cannot be opened because it has not been defined correctly, or because it has been closed by operator intervention.
- The connection manager transaction, or the user running it, does not have the necessary level of authority to access the file.

System Action: The requested operation is not performed.

User Response: See the *CICS for MVS/ESA Application Programming Reference* for the meaning of the EIBRESP value, and take appropriate action.

Destination: CWBO

Module: DFHWBC09

XMEOUT Parameters: *date, time, applid, tranid, filename, eibresp*

DFHWB1511 *date time applid tranid* **The CICS Web Interface connection manager has detected a logic error accessing the CICS Web Interface data set, CICS file** *filename***.**

Explanation: The connection manager received an unexpected error when accessing the CICS Web Interface data set, CICS file *filename*. This is a logic error. The connection manager has received an

unexpected response from CICS following an EXEC CICS command.

System Action: A system dump is taken. The requested operation is not performed. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC09

XMEOUT Parameters: *date, time, applid, tranid, filename*

DFHWB1512 *date time applid tranid* **The CICS Web Interface connection manager cannot access the CICS Web Interface data set, CICS file** *filename***.**

Explanation: The connection manager could not access the CICS Web Interface data set, CICS file *filename*. The data set has been incorrectly defined to CICS for one of the following reasons:

- No file definition has been found for *filename*. The CICS Web Interface has therefore not been installed correctly.
- READ operations are not allowed on the file.
- The file has been disabled, either due to an incorrect data set definition, or due to operator intervention.
- The file cannot be opened because it has not been defined correctly, or because it has been closed by operator intervention.
- The connection manager transaction, or the user running the connection manager, does not have the authority necessary to access the file.

System Action: The message is displayed at the terminal.

User Response: Ensure that all the CEDA groups for the CICS Web Interface have been installed correctly.

Investigate whether the operator has changed the status of the file for any reason.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid, filename*

DFHWB1514 *date time applid tranid* **The CICS Web Interface connection manager has detected that the CICS Web Interface global work area does not have the expected length.**

Explanation: The connection manager detected that the length of the associated global work area is not correct.

System Action: A system dump is taken. The CICS Web Interface is disabled. It is not possible to enable the CICS Web Interface until the problem is resolved. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: Ensure that no user-written version of program DFHWBTRU is being used. Only the CICS supplied program can be used with the CICS Web Interface. Similarly, the CICS Web Interface supplied task-related user exit DFHWBTRU should be enabled and disabled only by the connection manager. It should not be necessary to enable or disable DFHWBTRU in any other way.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1515 *date time applid tranid* **The CICS Web Interface connection manager detected an error while accessing the CICS Web Interface data set, CICS file** *filename***. EIBRESP:** *eibresp***.**

Explanation: The connection manager could not access the CICS Web Interface data set, CICS file *filename*. An EXEC CICS READ was issued, but received the response *eibresp*. The error can occur for one of the following reasons:

- The file is defined as remote, and there is an error on the connection to the owning system.
- VSAM has returned an unexpected response to CICS.
- An I/O error occurred on the READ.

System Action: A system dump is taken. The requested operation is not performed. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: See the *CICS for MVS/ESA Application Programming Reference* for the meaning of the EIBRESP value, and take appropriate action.

Destination: CWBO

Module: DFHWBC0

XMEOUT Parameters: *date, time, applid, tranid, filename, eibresp*

DFHWB1516 *date time applid tranid* **The CICS Web Interface connection manager cannot access the Interface definition record in the CICS Web Browser Interface data set, CICS file** *filename***.**

Explanation: The connection manager found that the CICS Web Interface definition record is missing from the CICS Web Interface data set, CICS file *filename*, while processing a request to update this record.

System Action: A system dump is taken. The connection manager panel is redisplayed. The CICS Web Interface definition record cannot be updated. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: Investigate why this record is missing. Create a new definition record using the connection manager.

Destination: CWBO

Module: DFHWBC0

XMEOUT Parameters: *date, time, applid, tranid, filename*

DFHWB1518 *date time applid tranid* **The CICS Web Interface connection manager cannot find the global work area.**

Explanation: The connection manager cannot access its global work area.

System Action: A system dump is taken. The connection manager continues, but the CICS Web Interface cannot be enabled. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: End the connection manager. Ensure that all the CEDA groups containing the CICS Web Interface definitions have been correctly installed. Then try running the connection manager again.

Investigate whether the operator has disabled the task-related user exit DFHWBTRU.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1519 *date time applid tranid* **The CICS Web Interface connection manager cannot find the task-related user exit.**

Explanation: The connection manager cannot access its task-related user exit because DFHWBTRU is:

- Not defined to CICS
- Not in the CICS load library
- Disabled

System Action: A system dump is taken. The CICS Web Interface is disabled. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: End the connection manager. Ensure that all the CEDA groups containing the CICS Web Interface definitions have been installed correctly. Then try running the connection manager again.

If the CICS Web Interface has been correctly installed, check that the operator has not disabled DFHWBTRU.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1520 *date time applid tranid* **The CICS Web Interface connection manager is not authorized to access its task-related user exit. EIBRESP2:** *eibresp2***.**

Explanation: The connection manager used EXEC CICS EXTRACT EXIT to find the task-related user exit, but received a NOTAUTH response.

System Action: A system dump is taken. The CICS Web Interface is disabled. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: Use the EIBRESP2 value to identify the problem.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid, eibresp2*

DFHWB1521 *date time applid tranid* **The CICS Web Interface connection manager cannot access its task-related user exit.**

Explanation: The connection manager cannot access the task-related user exit. It received an unexpected response to an EXEC CICS EXTRACT EXIT call.

System Action: A system dump is taken. The CICS Web Interface is disabled. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1522 *date time applid tranid* **The CICS Web Interface connection manager has been started against an invalid terminal.**

Explanation: The connection manager has been started against a terminal that is not supported, for example, an LUTYPE6 terminal.

System Action: The connection manager abends with abend code AWBX.

User Response: Start the connection manager against a valid terminal. See "Chapter 5. Configuration with the connection manager" on page 55 for further guidance on starting the connection manager.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1523 *date time applid tranid* **The CICS Web Interface cannot be enabled because the connection manager cannot access the task-related user exit DFHWBTRU.**

Explanation: The connection manager could not enable the CICS Web Interface because an error occurred accessing the task related user exit DFHWBTRU.

System Action: A system dump is taken. This instance of connection manager can only be used to inquire on, or update the CICS Web Interface data set. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response:

See the CWBO transient data queue for messages indicating the nature of the error, and take the appropriate action. Then restart the connection manager transaction CWBC and select the enable option again.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1524 *date time applid tranid* **The CICS Web Interface cannot be enabled because the server controller is already running.**

Explanation: The connection manager detected that the task-related user exit DFHWBTRU is disabled, but the server controller transaction CWBM is still running.

System Action: A system dump is taken. This instance of connection manager can only be used to inquire on, or update, the CICS Web Interface data set. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: Investigate why the last attempt to disable the CICS Web Interface did not complete successfully. Investigate the possibility of operator intervention.

Once you have established that it is safe to continue, use CEMT SET TASK or EXEC CICS SET TASK to purge the server controller. Then run the connection manager again to enable the CICS Web Interface.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1525 *date time applid tranid* **The CICS Web Interface connection manager received an unexpected response from CICS.**

Explanation: The connection manager received an unexpected response to a CICS command. This is a logic error.

System Action: A system dump is taken. Processing continues. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1526 *date time applid tranid* **The CICS Web Interface connection manager found that the task-related user exit is enabled, but the server controller is not running.**

Explanation: The connection manager has detected that the task-related user exit DFHWBTRU is enabled, but the server controller is not running. This means that the CICS Web Interface is in an indeterminate state.

System Action: This instance of connection manager can only be used to inquire on, or update, the CICS Web Interface data set.

User Response: Investigate whether the previous attempt to disable the CICS Web Interface completed successfully. Alternatively, the server controller task may have been forcepurged by the operator.

Having established that it is safe to continue, rerun the connection manager and try to re-enable the CICS Web Interface.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1531 *date time applid tranid* **The CICS Web Interface connection manager detected an error while accessing the CICS Web Interface data set, CICS file** *filename***. EIBRESP:** *eibresp***.**

Explanation: The connection manager could not access the CICS Web Interface data set, CICS file *filename*. An EXEC CICS WRITE was issued, but received the response *eibresp*. The error can occur for one of the following reasons:

- No file definition has been found for the file. This means that the CICS Web Interface has not been installed correctly.
- Write operations are not allowed. This means that the CICS Web Interface has not been installed correctly.
- The file is DISABLED. This is due either to an incorrect file definition, or to operator intervention.
- The file is NOTOPEN, This is due either to incorrect file definition or to operator intervention.
- Write operations are not authorized. This means that security has not been set up correctly.

System Action: The requested operation is not performed.

User Response: See the *CICS for MVS/ESA Application Programming Reference* for the meaning of the EIBRESP value, and take appropriate action.

Destination: CWBO

Module: DFHWBC0

XMEOUT Parameters: *date, time, applid, tranid, filename, eibresp*

DFHWB1532 *date time applid tranid* **The CICS Web connection manager detected an error while accessing the CICS Web Interface data set, CICS file** *filename***. EIBRESP:** *eibresp***.**

Explanation: The connection manager could not access the CICS Web Interface data set, CICS file *filename*. An EXEC CICS WRITE was issued, but received the response *eibresp*. The error can occur for one of the following reasons:

- The file is defined as remote, and there is an error on the connection to the owning system.
- VSAM has returned an unexpected response to CICS.
- An I/O error occurred on the WRITE.
- There is insufficient space available on the DASD device containing the data set.

System Action: A system dump is taken. The requested operation is not performed. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: See the *CICS for MVS/ESA Application Programming Reference* for the meaning of the EIBRESP value, and take appropriate action.

Destination: CWBO

Module: DFHWBC0

XMEOUT Parameters: *date, time, applid, tranid, filename, eibresp*

DFHWB1533 *date time applid tranid* **The CICS Web Interface connection manager has detected a logic error while accessing the CICS Web Interface data set, CICS file** *filename***.**

Explanation: The connection manager used EXEC CICS WRITE to update the CICS Web Interface data set, but received an unexpected response. This is a logic error.

System Action: A system dump is taken. The requested operation is not performed. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC0

XMEOUT Parameters: *date, time, applid, tranid, filename*

DFHWB1540 *date time applid tranid* **The CICS Web Interface connection manager detected a logic error.**

Explanation: The connection manager received an unexpected response from CICS following an EXEC CICS command.

System Action: A system dump is taken. The connection manager abends with abend code AWBV. The other components of the CICS Web Interface continue. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1541 *date time applid tranid* **The CICS Web Interface connection manager detected a logic error.**

Explanation: The connection manager received an unexpected response from CICS following an EXEC CICS command.

System Action: A system dump is taken. The enable attempt is abandoned. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1548 *date time applid tranid* **The CICS Web Interface connection manager detected an error attempting to retrieve fast path data. EIBRESP:** *eibresp***.**

Explanation: The connection manager was attempting to retrieve fast path commands specified when it was initiated from a terminal. The connection manager issued an EXEC CICS RECEIVE command, but received a response in field *eibresp*.

System Action: A system dump is taken. The connection manager continues but any fast path commands are ignored. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid, eibresp*

DFHWB1549 *date time applid tranid* **The CICS Web Interface connection manager received an error response while registering with CICS for problem determination.**

Explanation: The connection manager received an unexpected response from CICS when attempting to register for problem determination.

System Action: A system dump is taken. CICS feature tracing and dump formatting cannot be used for the CICS Web Interface. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC01

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1550 *date time applid tranid* **The CICS Web Interface connection manager received an error response while registering with CICS for problem determination.**

Explanation: The connection manager received an unexpected response from CICS when attempting to register for problem determination.

System Action: A system dump is taken. The enable attempt is abandoned. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1554 *date time applid tranid* **The CICS Web Interface connection manager is not authorized to use the CICS SPI.**

Explanation: The connection manager has not been defined with the authorization necessary to execute CICS system programming interface commands. It cannot function without this authorization.

System Action: A system dump is taken. The enable attempt is abandoned.

User Response: Message DFHME0116 is normally produced containing the symptom string for this problem. Redefine the connection manager transaction and its associated program DFHWBC00 with the level of security necessary to use the CICS SPI.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1555 *date time applid tranid* **The CICS Web Interface connection manager is not authorized to use the program DFHWBTRU.**

Explanation: The connection manager used the EXEC CICS ENABLE PROGRAM command for DFHWBTRU, but it has not been defined with the authorization necessary to use DFHWBTRU. It cannot function without this authorization.

System Action: A system dump is taken. The enable attempt is abandoned. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: Redefine the connection manager and its associated programs with the level of security necessary to use the CICS Web Interface supplied task-related user exit DFHWBTRU.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1556 *date time applid tranid* **The CICS Web Interface connection manager has detected an internal error during enable processing.**

Explanation: An internal error detected by the connection manager during enable processing has prevented the CICS Web Interface from being enabled.

System Action: A system dump is taken. The enable attempt is abandoned. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1564 *date time applid tranid* **The CICS Web Interface could not be enabled due to an internal error while starting the server controller. Host IP address:** *hostaddr***.**

Explanation: The connection manager attempted to start the server controller by issuing an EXEC CICS START command, but could not identify the response.

System Action: A system dump is taken. The enable attempt is abandoned. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid, hostaddr*

DFHWB1565 *date time applid tranid* **The CICS Web Interface cannot be enabled because the connection manager is not authorized to start the server controller. EIBRESP:** *eibresp***. Host IP address:** *hostaddr***.**

Explanation: The connection manager attempted to start the server controller by issuing an EXEC CICS START command, but the NOTAUTH response was returned.

System Action: A system dump is taken. The enable attempt is abandoned. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: See the *CICS for MVS/ESA System Programming Reference* for the meaning of the value returned in *eibresp*. Use CEDA to ensure that the resource definitions for the CICS Web Interface supplied programs and transactions have been defined with the correct levels of security. The connection manager must have the correct level of authority to start the server controller before the CICS Web Interface to be enabled.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid, eibresp, hostaddr*

DFHWB1566 *date time applid tranid* **The CICS Web Interface cannot be enabled due to an error starting the server controller. EIBRESP:** *eibresp***. Host IP address:** *hostaddr***.**

Explanation: The connection manager attempted to start the server controller by issuing an EXEC CICS START command, but the TRANSIDERR response was returned.

See the *CICS for MVS/ESA System Programming Reference* for the meaning of the value returned in *eibresp*.

System Action: A system dump is taken. The enable attempt is abandoned. Message DFHME0116 is

normally produced containing the symptom string for this problem.

User Response: Use CEDA to ensure that the resource definitions for the server controller supplied programs and transactions have been defined and installed correctly.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid, eibresp, hostaddr*

DFHWB1567 *date time applid tranid* **The CICS Web Interface could not be enabled due to a security error starting the server controller. User ID** *userid* **is unknown. Host IP address:** *hostaddr***.**

Explanation: The connection manager attempted to start the server controller by issuing an EXEC CICS START USERID command, but the USERIDERR response was returned.

The user ID specified for the server controller is not known to the external security manager.

System Action: The enable attempt is abandoned.

User Response: Ensure that a valid user ID is specified for CWBM Userid.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid, userid, hostaddr*

DFHWB1568 *date time applid tranid* **The CICS Web Interface could not be enabled due to a security error starting the server controller. Host IP address:** *hostaddr***.**

Explanation: The connection manager attempted to start the server controller by issuing an EXEC CICS START USERID command, but the USERIDERR response was returned.

The external security manager cannot validate the user ID specified for the server controller.

System Action: A system dump is taken. The enable attempt is abandoned. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: Investigate the reason why the external security manager cannot perform this request.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid, hostaddr*

DFHWB1575 *date time applid tranid* **The CICS Web Interface could not be enabled due to an internal error starting the server controller. Host IP address:** *hostaddr***.**

Explanation: The connection manager attempted to start the server controller by issuing an EXEC CICS START command, but received an unexpected response.

System Action: A system dump is taken. The enable attempt is abandoned. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid, hostaddr*

DFHWB1576 *date time applid tranid* **The CICS Web Interface could not be enabled due to an internal error starting the server controller. EIBRESP:** *eibresp***. Host IP address:** *hostaddr***.**

Explanation: The connection manager attempted to start the server controller by issuing an EXEC CICS START command, but the INVREQ response was returned.

System Action: A system dump is taken. The enable attempt is abandoned. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid, eibresp, hostaddr*

DFHWB1577 *date time applid tranid* **The CICS Web Interface connection manager cannot access its task-related user exit DFHWBTRU.**

Explanation: The connection manager was unable to access its task-related user exit DFHWBTRU during enable processing.

System Action: A system dump is taken. The enable attempt is abandoned. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: Check that the task-related user exit has not been disabled by operator intervention. See the associated diagnostics issued by CICS for problem determination.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1580 *date time applid tranid* **The CICS Web Interface connection manager cannot establish whether security is active or obtain the default CICS user ID. EIBRESP:** *eibresp***.**

Explanation: The connection manager was unable to retrieve CICS status information, and therefore cannot establish whether security is active, or obtain the default CICS user ID.

An EXEC CICS INQUIRE SYSTEM was issued but received the response shown in the message.

System Action: Processing continues under the assumption that there is no security active.

Panel DFHWB02 is displayed with no user ID in field CWBM Userid, unless a user ID was saved in the CICS Web Interface data set.

User Response: Ensure that the connection manager has the correct level of security to use CICS system programming interface commands.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid, eibresp*

DFHWB1581 *date time applid tranid* **The CICS Web Interface connection manager detected an internal error while accessing the CICS Web Interface data set, CICS file** *filename***.**

Explanation: The connection manager has detected an internal error while accessing the CICS Web Interface data set.

System Action: A system dump is taken. The panel is redisplayed. No records can be updated. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC03

XMEOUT Parameters: *date, time, applid, tranid, filename*

DFHWB1582 *date time applid tranid* **The CICS Web Interface connection manager detected an internal error while accessing the CICS Web Interface data set, CICS file** *filename***.**

Explanation: The connection manager has detected an internal error while accessing the CICS Web Interface data set.

System Action: A system dump is taken. The connection manager panel is redisplayed. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC03

XMEOUT Parameters: *date, time, applid, tranid, filename*

DFHWB1596 *date time applid tranid* **The CICS Web Interface connection manager cannot continue enable processing because it cannot determine the status of the CICS Web Interface.**

Explanation: The connection manager was trying to enable the CICS Web Interface, but detected an invalid global work area address, or found that it was already enabled.

System Action: The enable attempt is abandoned.

User Response: Investigate whether the CICS Web Interface has been disabled. Investigate whether operator command have been issued against the task-related user DFHWBTRU.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1607 *date time applid tranid* **The CICS Web Interface connection manager has detected an error when attempting to disable the CICS Web Interface TRUE (DFHWBTRU).**

Explanation: The connection manager detected an error attempting to disable the task-related user exit (TRUE). This is during backout of enable processing initiated by the connection manager in response to a failed enable request. The TRUE might already be disabled as a result of operator intervention. Alternatively, this could be a result of problems with the CICS Web Interface or with CICS.

System Action: The CICS Web Interface continues backout of enable processing.

User Response: See any associated messages reporting the failure of the enable request. Take steps to prevent operator interference with the TRUE.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1608 *date time applid tranid* **The CICS Web Interface connection manager has detected an internal error during backout of enable processing.**

Explanation: The connection manager has detected an internal error while attempting to disable the task-related user exit (TRUE). This occurred during backout of enable processing initiated by the connection manager in response to a failed enable request. This could be due to problems with the CICS Web Interface or with CICS itself.

System Action: The CICS Web Interface continues backout of enable processing.

User Response: See any associated messages reporting the failure of the enable request. If you cannot identify an underlying problem, you will need further assistance from IBM. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1609 *date time applid tranid* **The CICS Web connection manager is not authorized to disable the task-related user exit (DFHWBTRU) during backout of enable processing.**

Explanation: The connection manager transaction does not have the authority necessary to use the CICS system programming interface and cannot disable its task related user exit (TRUE). This occurred during backout of enable processing initiated by the connection manager in response to a failed enable request. This could be due to problems with the CICS Web Interface or to problems with CICS itself.

System Action: The CICS Web Interface continues backout of enable processing.

User Response: See any associated messages reporting the failure of the enable request. If you cannot identify an underlying problem, you will need further assistance from IBM. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1610 *date time applid tranid* **The CICS Web connection manager is not authorized to disable the task-related user exit (DFHWBTRU) during backout of enable processing.**

Explanation: The connection manager transaction does not have the authority necessary to disable its task-related user exit (TRUE). This occurred during backout of enable processing initiated by the connection manager in response to a failed enable request. This could be due to problems with the CICS Web Interface or to problems with CICS itself.

System Action: The CICS Web Interface continues backout of enable processing.

User Response: See any associated messages reporting the failure of the enable request. If you cannot identify an underlying problem, you will need further assistance from IBM. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed.

Destination: CWBO

Module: DFHWBC42

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1650 *date time applid tranid* **The CICS Web Interface connection manager found that the interface is disabled. Requests to disable the interface are ignored.**

Explanation: A request has been made to disable the CICS Web Interface, but it is already disabled, or in the process of being disabled.

System Action: The request is ignored. The connection manager panel is redisplayed.

User Response: Request another option.

Destination: Terminal End User

Module: DFHWBC04

DFHWB1651 *date time applid tranid* **The CICS Web Interface connection manager detected a logic error.**

Explanation: The connection manager has received an unexpected response from CICS following an EXEC CICS command.

System Action: A system dump is taken. The connection manager abends with abend code AWBV. The rest of the CICS Web Interface continues. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: You need further assistance from IBM to resolve this problem. See "Chapter 7. Problem determination" on page 69, and part 4 of the *CICS for MVS/ESA Problem Determination Guide* for guidance on how to proceed. Report the details of the symptom string given in message DFHME0116.

Destination: CWBO

Module: DFHWBC04

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1900 *date time applid tranid* **The CICS Web Interface connection manager could not find the global work area.**

Explanation: The connection manager could not find the global work area. The task related user exit DFHWBTRU has been defined incorrectly.

System Action: A system dump is taken. The CICS Web Interface remains disabled. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: Ensure that all the CEDA groups for the CICS Web Interface have been installed correctly, then try to enable the CICS Web Interface again.

Investigate whether the operator has disabled DFHWBTRU.

Destination: CWBO

Module: DFHWBC0B

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1901 *date time applid tranid* **The CICS Web Interface connection manager could not find the task-related user exit program DFHWBTRU.**

Explanation: The connection manager cannot find the task-related user exit, DFHWBTRU, for one of the following reasons:

- DFHWBTRU has not been defined to CICS.
- DFHWBTRU is not in the CICS load library.
- DFHWBTRU has been disabled.

System Action: A system dump is taken. The CICS Web Interface remains disabled. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: Ensure that all the CEDA groups for the CICS Web Interface have been installed correctly, then try the enable request again.

Destination: CWBO

Module: DFHWBC0B

XMEOUT Parameters: *date, time, applid, tranid*

DFHWB1902 *date time applid tranid* **The CICS Web Interface connection manager does not have sufficient authority to issue the EXEC CICS EXTRACT EXIT command. EIBRESP2:** *eibresp2***.**

Explanation: The connection manager does not have the authority necessary to issue the privileged EXEC CICS EXTRACT EXIT command. It cannot function without this authority.

System Action: A system dump is taken. The connection manager abends with abend code AWBZ. Message DFHME0116 is normally produced containing the symptom string for this problem.

User Response: Use the EIBRESP2 value to identify the problem. Ensure that the connection manager and its associated program DFHWBC00 have the authority necessary to issue the EXEC CICS EXTRACT EXIT command for the CICS Web Interface task related user exit DFHWBTRU.

Destination: CWBO

Module: DFHWBC0B

XMEOUT Parameters: *date, time, applid, tranid, eibresp2*

Index

Numerics

200 response
  HTTP response 4
302 response
  HTTP response 67
400 response
  business logic interface 91
  HTTP response 31, 37, 39
401 response
  business logic interface 91
403 response
  business logic interface 91
  HTTP response 36
404 response
  business logic interface 91
500 response
  business logic interface 91
  HTTP response 37
501 response
  HTTP response 37, 39
503 response
  business logic interface 92

A

absolute path in URL 3
Accept-Encoding HTTP header 50
Accept HTTP header 50
Accept-Language HTTP header 50
alias 4, 66
analyzer 5
  default 21, 32
  designing and coding 19
  programming reference 26
  security sample 66
Authorization HTTP header 50

B

Backlog panel field 17, 61
business logic 2, 85
business logic interface 86
  programming reference 87

C

CGI (common gateway interface) 2
Charge-To HTTP header 50
CICS Internet Gateway 2
CICS program
  designing and coding 41
CICS system initialization parameters
  XPCT 65
  XTRAN 65
  XUSER 65
CICS Web Interface data set 5, 14, 63
common gateway interface (CGI) 2
connection manager 4, 55, 65
connection manager panels
  DFHWB01 58
connection manager panels *(continued)*
  DFHWB02 17, 60
  DFHWB04 59
  DFHWB06 62
  DFHWB12 17, 63
  DFHWB16 64
CONTENT_LANGUAGE environment variable 50
Content-Language HTTP header 21, 50
CONTENT_LENGTH environment variable 50
Content-Length HTTP header 50
CONTENT_TYPE environment variable 50
Content-Type HTTP header 21, 50
conversation token 51
converter 5
  designing and coding 23
  programming reference 32
  security sample 66
CRLF 3
CWBA 4, 13
CWBC 4, 56
CWBM 4
CWBM Userid panel field 59
CWBO TD destination 57

D

data conversion 7, 15
decode_client_address field 34
decode_client_address_string field 34
Decode converter function
  designing and coding 23
  programming reference 34
decode_data_ptr field 34
decode_eyecatcher field 34
decode_function field 34
decode_http_version_length field 35
decode_http_version_ptr field 35
decode_input_data_len field 23, 35
decode_method_length field 35
decode_method_ptr field 35
decode_output_data_len field 35, 38
decode_reason field 35
decode_request_header_length field 35
decode_request_header_ptr field 35
decode_resource_length field 35
decode_resource_ptr field 36
decode_response field 36
decode_server_program field 30, 36
decode_user_data_length field 36
decode_user_data_ptr field 36
decode_user_token field 30, 36, 39, 91
DFH$WB1A 16, 42
DFH$WBSA 66
DFH$WBSC 66
DFH$WBSN 66
DFH$WBSN RDO group 13
DFH$WBSR 51
DFH$WBST 51
DFHCNV table 15, 61
DFHHTML DD name 15
DFHIWBD job 12
DFHIWBL job 12
DFHWBA 13
DFHWBA1 87
DFHWBADX 21, 32
DFHWBCD file definition 14
DFHWBENV 49
DFHWBHH conversion template 15
DFHWBHH conversion template name 15
DFHWBJ41 job 13
DFHWBPA 53
DFHWBTL 42
DFHWBUD conversion template 15
DFHWBUD conversion template name 16, 22
DFHWEB RDO group 13
DFHWEBF RDO group 13
double-byte character set (DBCS) 16
DPL subset 42

E

EDF 82
Enable DNS panel field 61
Encode converter function
  designing and coding 25
  programming reference 38
encode_data_ptr field 38
encode_eyecatcher field 38
encode_function field 38
encode_input_data_len field 38
encode_reason field 38
encode_response field 38
encode_user_token field 36, 39
ENTER TRACENUM command 82
environment variables program 6, 49
EXCI 2
external presentation interface 2

F

From HTTP header 50

H

HANDLE ABEND command 83
hidden field in HTML form 52
host name in URL 3
HTML 2
HTML form 3
HTML template manager 6, 42, 65
  programming reference 43
  setting up a PDS 14
HTTP 1
HTTP_ACCEPT_ENCODING environment variable 50
HTTP_ACCEPT environment variable 50

HTTP_ACCEPT_LANGUAGE
environment variable 50
HTTP_AUTHORIZATION environment
variable 50
HTTP_CHARGE_TO environment
variable 50
HTTP_FROM environment variable 50
HTTP_IF_MODIFIED_SINCE
environment variable 50
HTTP method 3
HTTP_PRAGMA environment
variable 50
HTTP_REFERER environment
variable 50
HTTP request 3
HTTP request header 3
Accept 50
Accept-Encoding 50
Accept-Language 50
Authorization 50
Charge-To 50
Content-Language 21, 50
Content-Length 50
Content-Type 21, 50
From 50
If-Modified-Since 50
Pragma 50
Referer 50
User-Agent 50
HTTP response 4
HTTP response codes 4
HTTP response header 4
HTTP_USER_AGENT environment
variable 50
HTTP user data 3
HTTP version 3
hypertext markup language 2
hypertext transfer protocol 1

I

IBM Internet Connection Server 2
If-Modified-Since HTTP header 50
installing the CICS Web Interface 12
IPCS VERBEXIT 82
ISO 3316 language code 50
ISO 639 country code 50
ISO 8859-1 character set 15

L

Latin-1 character set 15

M

messages and codes 71

N

name server 17
non-HTTP requests 9

P

parser program 6, 53
PLT program 56, 65
PLTPIUSR system initialization
parameter 65
port number in URL 3
Port Number panel field 61
Pragma HTTP header 50
presentation logic 2, 85

Q

QR TCB 70
QUERY_STRING environment
variable 50
query string in URL 3, 52

R

Referer HTTP header 50
REMOTE_ADDR environment
variable 51
REMOTE_HOST environment
variable 51
REMOTE_USER environment
variable 51
REQUEST_METHOD environment
variable 51
RP TCB 70

S

samples
application 16, 42
security analyzer 66
security converter 66
sign-on program 66
state management program 51
security 65
security analyzer 66
security converter 66
server controller 4, 66
SERVER_NAME environment
variable 51
SERVER_PORT environment variable 51
SERVER_PROTOCOL environment
variable 51
SERVER_SOFTWARE environment
variable 51
session token 52
shutdown 62, 63
sign-on sample program 66
state management sample program 51,
66
symbol list 47
symbols in an HTML template 47
SYSTCPD DD name 17

T

task control blocks 70
TCP/IP port in URL 3
tools
environment variables program 6, 49
HTML template manager 6, 42, 65
parser program 6, 53
trace points 72
two-tier programming model 2, 85

U

Unescape panel field 61
unescaping considerations 54
uniform resource locator 3
URL 3
absolute path 3
host name 3
port number 3
query string 3
URP_DISASTER response
in analyzer 31
in Decode 37
in Encode 39
URP_EXCEPTION response
in analyzer 31
in Decode 36
in Encode 39
URP_INVALID response
in analyzer 31
in Decode 37
in Encode 39
URP_OK response
in analyzer 31
in Decode 36
in Encode 39
URP_RECEIVE_OUTSTANDING reason
code 31
User-Agent HTTP header 50
user data 3
user-replaceable program 2, 5, 19

V

VERBEXIT 82

W

wba1_blip_eyecatcher field 89
wba1_client_address field 89
wba1_client_address_string field 89
wba1_converter_program_name field 89
wba1_data_ptr field 89
wba1_header_length field 89
wba1_header_offset field 89
wba1_http_version_length field 89
wba1_http_version_offset field 90
wba1_input_data_length field 90
wba1_method_length field 90
wba1_method_offset field 90
wba1_outdata_ptr field 90
wba1_resource_length field 90
wba1_resource_offset field 90
wba1_response field 90
wba1_server_program_name field 90
wba1_user_data_length field 90
wba1_user_data_offset field 90
wba1_user_token field 91
wbra_alias_termid field 28, 82
wbra_alias_tranid field 28
wbra_client_ip_address field 28
wbra_content_length field 28
wbra_converter_program field 28
wbra_dfhcnv_key field 28
wbra_eyecatcher field 28
wbra_function field 28
wbra_http_header_ptr_field 28
wbra_http_version_length field 29
wbra_http_version_ptr field 29
wbra_method_length field 29
wbra_method_ptr field 29

wbra_reason field 29
wbra_request_header_length field 29
wbra_request_header_ptr field 29
wbra_request_type field 29
wbra_resource_length 28
wbra_resource_length field 29
wbra_resource_ptr field 29
wbra_response field 29
wbra_server_ip_address field 30
wbra_server_program field 30, 36
wbra_unescape field 30
wbra_user_data_length field 30
wbra_user_data_ptr field 30
wbra_user_token field 30, 36
wbra_userid field 31, 66
wbtl_connect_token field 45
WBTL_DISASTER response 47
WBTL_EXCEPTION response 46
WBTL_FREEMAIN_ERROR reason 47
wbtl_function field 44
WBTL_GETMAIN_ERROR reason 47
wbtl_html_buffer_len field 46
wbtl_html_buffer_ptr field 46
WBTL_INVALID_BUFFER_PTR reason 47
WBTL_INVALID_FUNCTION reason 47
WBTL_INVALID response 47
WBTL_INVALID_SYMBOL_LIST reason 47
WBTL_INVALID_TOKEN reason 47
WBTL_OK response 46
wbtl_reason field 45
wbtl_response field 45
wbtl_symbol_list_len field 46
wbtl_symbol_list_ptr field 46
wbtl_template_abstime field 45
wbtl_template_buffer_len field 46
wbtl_template_buffer_ptr field 45
wbtl_template_name field 45
WBTL_TEMPLATE_NOT_FOUND reason 47
WBTL_TEMPLATE_TRUNCATED reason 47
wbtl_version_no field 44
WRITEQ TD command 83

Program Number: 5655-018

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC33-1892-01