ASG-Web Enabler™
System Administrator's Guide

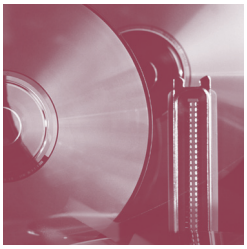
Version: 2.5.1

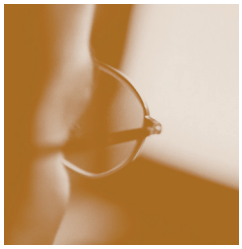
Publication Number: WBR2100-251

Publication Date: November 2001

The information contained herein is the confidential and proprietary information of Allen Systems Group, Inc. Unauthorized use of this information and disclosure to third parties is expressly prohibited. This technical publication may not be reproduced in whole or in part, by any means, without the express written consent of Allen Systems Group, Inc.

© 1998-2001 Allen Systems Group, Inc. All rights reserved.
All names and products contained herein are the trademarks or registered trademarks of their respective holders.





ASG Worldwide Headquarters Naples, Florida USA | asg.com
1333 Third Avenue South, Naples, Florida 34102 USA Tel: 941.435.2200 Fax: 941.263.3692 Toll Free: 1.800.932.5536

ASG Documentation/Product Enhancement Fax Form

Please FAX comments regarding ASG products and/or documentation to (941) 263-3692.

Company Name	Telephone Number	Site ID	Contact name

Product Name/Publication	Version #	Publication Date
Product:		
Publication:		
Tape VOLSER:		

Enhancement Request:

© 2001 Allen Systems Group, Inc.
All names and products are trademarks or registered trademarks of their respective holders.

ASG Support Numbers

ASG provides support throughout the world to resolve questions or problems regarding installation, operation, or use of our products. We provide all levels of support during normal business hours and emergency support during non-business hours. To expedite response time, please follow these procedures.

Please have this information ready:

- Product name, version number, and release number
- List of any fixes currently applied
- Any alphanumeric error codes or messages written precisely or displayed
- A description of the specific steps that immediately preceded the problem
- The severity code (ASG Support uses an escalated severity system to prioritize service to our clients. The severity codes and their meanings are listed below.)
- Verify whether you received an ASG Service Pack for this product. It may include information to help you resolve questions regarding installation of this ASG product. The Service Pack instructions are in a text file on the distribution media included with the Service Pack.

If You Receive a Voice Mail Message:

1. Follow the instructions to report a production-down or critical problem.
2. Leave a detailed message including your name and phone number. A Support representative will be paged and will return your call as soon as possible.
3. Please have the information described above ready for when you are contacted by the Support representative.

Severity Codes and Expected Support Response Times

Severity	Meaning	Expected Support Response Time
1	Production down, critical situation	Within 30 minutes
2	Major component of product disabled	Within 2 hours
3	Problem with the product, but customer has work-around solution	Within 4 hours
4	"How-to" questions and enhancement requests	Within 4 hours

ASG provides software products that run in a number of third-party vendor environments. Support for all non-ASG products is the responsibility of the respective vendor. In the event a vendor discontinues support for a hardware and/or software product, ASG cannot be held responsible for problems arising from the use of that unsupported version.

Business Hours Support

Your Location	Phone	Fax	E-mail
United States and Canada	800.354.3578 1.941.435.2201 **Secondary Numbers:** 800.227.7774 800.525.7775	941.263.2883	support@asg.com
Australia	61.2.9460.0411	61.2.9460.0280	support.au@asg.com
England	44.1727.736305	44.1727.812018	support.uk@asg.com
France	33.141.028590	33.141.028589	support.fr@asg.com
Germany	49.89.45716.300	49.89.45716.400	support.de@asg.com
Singapore	65.224.3080	65.224.8516	support.sg@asg.com
All other countries:	1.941.435.2201		support@asg.com

Non-Business Hours - Emergency Support

Your Location	Phone	Your Location	Phone
United States and Canada	800.354.3578 1.941.435.2201 **Secondary Numbers:** 800.227.7774 800.525.7775 **Fax:** 941.263.2883		
Asia	011.65.224.3080	**Japan/Telecom**	0041.800.9932.5536
Australia	0011.800.9932.5536	**New Zealand**	00.800.9932.5536
Denmark	00.800.9932.5536	**South Korea**	001.800.9932.5536
France	00.800.9932.5536	**Sweden/Telia**	009.800.9932.5536
Germany	00.800.9932.5536	**Switzerland**	00.800.9932.5536
Hong Kong	001.800.9932.5536	**Thailand**	001.800.9932.5536
Ireland	00.800.9932.5536	**United Kingdom**	00.800.9932.5536
Israel/Bezeq	014.800.9932.5536		
Japan/IDC	0061.800.9932.5536	**All other countries**	1.941.435.2201

ASG Web Site

Visit http://www.asg.com, ASG's World Wide Web site.

Submit all product and documentation suggestions to ASG's product management team at http://www.asg.com/products/suggestions.asp

If you do not have access to the web, FAX your suggestions to product management at (941) 263-3692. Please include your name, company, work phone, e-mail ID, and the name of the ASG product you are using. For documentation suggestions include the publication number located on the publication's front cover.

Contents

Preface iii

About this Publication iii

Related Publications iv

Publication Conventions iv

1 Introduction 1

2 Installing Web Enabler 3

Required Software 3

Your Installation Package 4

Installing the Delivered Files 4

MP-AID Objects 4

Web Enabler Unix Files 4

Installing Web Enabler 12

Servlet Files 13

3 Tailoring Web Enabler 17

Defining the Navigation Tree 18

Adjusting Your Repository Environment 19

Tailoring Dynamic HTML Generation 21

Adjusting HTML Layout 21

Multiselection 22

Search 22

Object-type-specific Settings 22

Generating RIM-specific HTML Forms 25

Using a Servlet Connection 26

Object-type-specific Search Dialogs 26

Search Dialogs 27

Object-type-independent Search Form 28

New Object Form 29

Edit Forms 29
Editing Form Display in non-MethodManager Environments 30
Editing Form Display in MethodManager Environments 31

Building Your Own Pop-up 31

Building Your Own HTML Documents 32

Building Your Own Dialog Forms 34

HTML/CSS Processing 35

Using the Import Function 35

Web Enabler Message Display 36

4 Replicating Repository Data 37

Introduction 37

Object Views 38
Tailored Print 38
Tailored Explore 39
Object Report 39
Object Impact Analysis 39
Object Information 40

Object Navigation 40
Cluster Navigation 40
List Selection 42

Diagram Navigation 43

5 National Language Support 45

Servlet NLS 48

Index 49

Preface

This *ASG-Web Enabler System Administrator's Guide* provides information for system administrators about installing and configuring ASG-Web Enabler (herein called Web Enabler). Web Enabler is a customizable graphical user interface (GUI) that allows end-user access to a remote repository.

Allen Systems Group, Inc. (ASG) provides professional support to resolve any questions or concerns regarding the installation or use of any ASG product. Telephone technical support is available around the world, 24 hours a day, 7 days a week.

ASG welcomes your comments, as a preferred or prospective customer, on this publication or on any ASG product.

About this Publication

This publication consists of these chapters:

- Chapter 1, "Introduction," explains the use and general features of Web Enabler.
- Chapter 2, "Installing Web Enabler," provides step-by-step installation instructions.
- Chapter 3, "Tailoring Web Enabler," describes the features available in Web Enabler and the steps you can use to tailor the GUI to your site's needs.
- Chapter 4, "Replicating Repository Data," details the ways you can provide repository data to your users.
- Chapter 5, "National Language Support," covers National Language Support and EBCDIC and ASCII conversion considerations.

Related Publications

The documentation library for ASG-Web Enabler consists of these publications (where *nn* represents the product version number):

- *ASG-Web Enabler User's Guide* (WBR0200-*nn*) provides information about using ASG-Web Enabler.
- *ASG-Web Enabler System Administrator's Guide* (WBR2100-*nn*) provides information for system administrators about installing and configuring ASG-Web Enabler.
- *ASG-Manager Products Server Facility User's Guide* (MSF0200-*nn*) provides information about the installation and use of the ASG-Manager Products Server Facility.

Note:

To obtain a specific version of a publication, contact the ASG Service Desk.

Publication Conventions

Allen Systems Group, Inc. uses these conventions in technical publications:

Convention	Represents
ALL CAPITALS (excluding UNIX and JAVA)	Directory, path, file, dataset, member, database, program, command, and parameter names.
Initial Capitals on Each Word	Window, field, field group, check box, button, panel (or screen), option names, and names of keys. A plus sign (+) is inserted for key combinations (e.g., Alt+Tab).
`lowercase italic monospace`	Information that you provide according to your particular situation. For example, you would replace *`filename`* with the actual name of the file.
`Monospace`	Characters you must type exactly as they are shown. Code, JCL, file listings, or command/statement syntax. Also used for denoting brief examples in a paragraph.
bold (exclusively for UNIX)	Identifies commands, field names, program names, file names, directory names, and any other items that are predefined by the system.
Vertical Separator Bar (|) with underline	Options available with the default value underlined (e.g., Y|N).

1 Introduction

Web Enabler is a graphical user interface (GUI) that allows end-user access to a remote repository. Using a set of software modules you can connect to the Manager Products repository via the Internet. Additionally, there is a GUI available which you can use with every Repository Information Model (RIM) at any time. This GUI is tailorable and you can implement most of the customization by setting some variables in the repository.

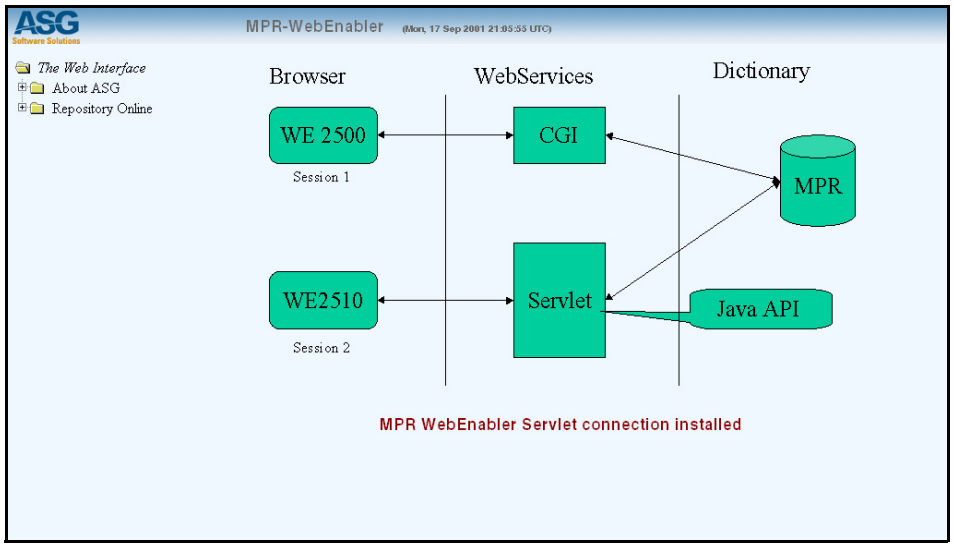
Note:

If you require specific functions that the standard Web Enabler GUI cannot process, contact ASG's Professional Services for information about customizing or expanding the functionality.

You can connect to the repository using these standard interfaces:

CGI. Use only with OS/390 POSIX environments.

Servlet. Makes use of the Jakarta-Tomcat classes, which means that every Tomcat enabled web server can be used such as Apache or WebSphere. Additionally, a workstation can make use of a local installation of Tomcat to connect to Manager Products. The Servlet connection to Manager Product Server Facility (MPSF) is done via TCP/IP and is in generally faster than the CGI interface.



Users can graphically search, report, list, analyze, update, delete, and create repository objects. Previous object selections can be saved for future use, or combined with other selections to support more elaborate searches. Users can view specific repository content diagrams, and the import and export of repository content is supported. An Expert mode is provided so that experienced users can enter lines of repository command syntax for specific processing requirements without leaving the web browser.

Web Enabler works with these web browsers:

- Microsoft Internet Explorer Version 5.0 or later
- Netscape Navigator Version 6.0 or later

No installation is necessary for the end-user; all content and functionality is supplied by Web Enabler components that reside on the web server and the repository. Refer to Chapter 2, "Installing Web Enabler," on page 3 for installation requirements.

2 Installing Web Enabler

This chapter provides step-by-step instructions for installing Web Enabler and contains these sections:

Required Software **3**

Your Installation Package **4**

Installing the Delivered Files **4**
MP-AID Objects 4
Web Enabler Unix Files 4
Installing Web Enabler 12
Servlet Files. 13

Required Software

These components must be present on your intranet in order for Web Enabler to function.

Host System

- IBM OS/390 Version 1.3.0 or later
- IBM ICS for OS/390 Version 2.1.0 or later

  Or

  IBM WebSphere Application Server for OS/390

  Or

  Lotus Domino Go WebServer for OS/390

  Or

  Any servlet enabled web server.
- ASG-MethodManager Version 2.5.1 or ASG-ControlManager (herein ControlManager) Version 2.5.1 with ASG-Manager Products Server Facility (MPSF).

Client System.

- Standard network PC with TCP/IP connection to an OS/390 web server
- Netscape Navigator (Version 4.0 or later)

  Or

  Microsoft Internet Explorer (Version 4.0 or later)
- A VRML viewer must be installed as a plug-in to display diagrams.

Your Installation Package

Your product package includes these items:

- ASG-Web Enabler on tape
- ASG-Manager Products Server Facility (MPSF) files for MVS on tape
- ASG-Web Enabler Servlet on compact disc (CD)

This publication provides installation instructions for Web Enabler only. Installation instructions for the other products are included in their respective publications.

Installing the Delivered Files

MP-AID Objects

All Web Enabler executive and command members are available after a basic ControlManager installation.

Web Enabler Unix Files

The Web Enabler Unix files are compressed with the UNIX **tar** command located on the tape, Label 27 MP.WEB.TAR.

Transfer and decompress the file to UNIX after loading it from the tape.

If the directory name **asg** is used for installation then you can start by using the ASG sample dictionary.

To extract the Web Enabler UNIX files

1. Load the file MP.WEB.TAR from the tape. See Figure 1 on page 5 for an example.

2. Log on to OMVS.

3. Change your working directory to the directory where you want to install Web Enabler.

4. Type this command to transfer the sequential file to HFS:

   `tso "oput 'the.sequential.file' '/xxx/we.tar' binary"`

   where *xxx* is your directory.

5. Type `tar -xfvo we.tar`

   The **tar** command extracts the Web Enabler UNIX files including new subdirectories.

Figure 1 • Example of JCL for Step 1

```
//LOADTAPE     JOB    ,,MSGCLASS=A,MSGLEVEL=(1,1),CLASS=A
     //*-----------------------------------------------------------------
     //*  COPY WEB ENABLER TAPE FILE TO DISK
     //*-----------------------------------------------------------------
     //*
     //LOAD1    EXEC PGM=IEBGENER
     //*        -----------------
     //SYSPRINT DD  SYSOUT=*
     //SYSUT1   DD  DISP=OLD,DSN=MP.WEB.TAR,
     //             UNIT=TAPE,VOL=(PRIVATE,RETAIN,SER=vol001),
     //             LABEL=(27,SL)
     //SYSUT2   DD  DISP=(NEW,CATLG),DSN=mp-WEB.TAR,
     //             UNIT=3390,VOL=SER=vvvvvv,SPACE=(TRK,(15),RLSE)
     //SYSIN    DD  DUMMY
     //*
```

Adjusting the CGI Environment

To adjust the environment so that all programs can be accessed

Note:

Your System Administrator for MVS, OMVS, and/or RACF functions should perform step 3 on page 6 through step 5 on page 7.

1. Log on to an OMVS environment.

2. Make the files and directories accessible to all users. These are examples of the commands:

```
chmod 755 /asg/cgi-bin
chmod 755 /asg/cgi-bin/*.*

chmod 755 /asg/html/static
chmod 755 /asg/html/static/*.*

chmod 755 /asg/html
chmod 755 /asg/html/*.*

chmod 755 /asg/html/gif
chmod 755 /asg/html/gif/*.*

chmod 755 /asg/html/sample
chmod 755 /asg/html/sample/*.*
```

3. Make the files and directories accessible to the web server by adding these lines to the configuration file of the ICSS (.../**etc/httpd.conf**):

```
AddType .wrl x-world/x-vrml ebcdic 0.5 # VRML virtual reality
AddType .htm text/html      ebcdic 1.0 # HTML on PCs
AddType .js  text/plain     ebcdic 0.5 # Java script
Pass    /asg/*      /asg/html/*
Exec    /asg-cgi/*  /asg/cgi-bin/*
```

Note:

It is not possible for a browser to access files and directories, if the line `Pass /asg/* asg/html*` in the **httpd.conf** is after the line `Pass /* ...`. This line closes the Pass definitions.

4 Add a STEPLIB assignment that contains the name of the MMR Load libraries to the start job of the web server. An example of the extension is in the file **httpdjob.asg**. These are samples of the assignment:

```
//STEPLIB  DD DSN=SYS1.IMW.SIMWMOD1,DISP=SHR
//         DD DSN=CUST.ASG.MPR.CMR2500.LOADLIB,DISP=SHR
```

Note:

Your LOADLIB *must* be APF authorized.

5 Adjust RACF for the web server's user ID and start job. The settings should be similar to normal online MMR-User; the web server and job user IDs need read-access to the load libraries, and the web server user ID needs update access for the corresponding repository filenames.

6 Make file allocations to repository files via settings in the file .../**cgi-bin/mmr_con.prf**. These allocations become active when you are connected to MethodManager (MMR) without the Server Facility; if the Server Facility is available, its file allocations are used. These are example entries:

```
DA_NAME.0=6
DA_NAME.1= 'CUST.ASG.CMR2400.MPAID'
DA_NAME.2= 'CUST.ASG.CMR2400.MPRBASE.MPAID'
DA_NAME.3= 'MP.CMR2400.DEMO.INDEX'
DA_NAME.4= 'MP.CMR2400.DEMO.SOURCE'
DA_NAME.5= 'MP.CMR2400.DEMO.DATA'
DA_NAME.6= 'MP.CMR2400.DEMO.ERROR'
DD_NAME.0=6
DD_NAME.1= 'MPAID'
DD_NAME.2= 'MPRBASE'
DD_NAME.3= 'DEMO'
DD_NAME.4= 'DEMOS'
DD_NAME.5= 'DEMOD'
DD_NAME.6= 'DEMOE'
```

7 Specify the name of the MMR server in the file .../**cgi-bin/mmr_con.cfg**. The server name is the name of the destination or local unit (LU) that was used to install the MMR server. For more details on server names, consult the *ASG-Manager Products Server Facility User's Guide*. This is an example specification statement:

```
SERVER= 'MPSERV02'
```

8 Specify the logon ID, password, and other logon commands to the configuration files **mmr_con.cfg** and **mmr_con.prf**.

Web Enabler tries to use an MMR server connection for each repository request. If a connection cannot be established, Web Enabler searches for a configuration file that contains other settings. The search path is set in the variable ALTFILENAME.

Define a server connection using these variables:

Name	Contents	Meaning
ALTFILENAME	delimited string	HFS-Name of another configuration file.
MPIOLOGON.0	integer	Size of the following MPIOLOGON array. If this value is set to 0, the Web Enabler user is prompted to enter a ControlManager logon ID and password.
MPIOLOGON.i	delimited string	Logon command.
SERVER	delimited string	The name of the MPSF server (APPC LU)

Define a non-server connection using these variables:

Name	Contents	Meaning
ALTFILENAME	delimited string	HFS-Name of another configuration file.
MPIOLOGON.0	integer	Size of the following MPIOLOGON array.
MPIOLOGON.i	delimited string	Logon command.
DA_NAME.0	integer	Size of the following DA_NAME array.
DA_NAME.i	delimited string	Physical name of ControlManger file to allocate. The name corresponds to the DD_NAME.
DD_NAME.0	integer	Size of the following DD_NAME array.
DD_NAME.i	delimited string	Logical name of ControlManger file to allocate. The name corresponds to the DA_NAME.

These are example definitions for the file **mmr_con.cfg**:

```
ALTFILENAME  = 'mmr_con.prf'
SERVER       = 'MPSF01'
MPIOLOGON.0  = 1
MPIOLOGON.1  = 'LOGON userid PASS password;'
```

These are example definitions for the file **mmm_con.prf**:

```
ALTFILENAME   = 'mmr_con_x.prf'
MPIOLOGON.0   = 4
MPIOLOGON.1   = 'LOGON userid PASS password;'
MPIOLOGON.2   = 'DICT SAMPLE;'
MPIOLOGON.3   = 'AU SAMPLE;'
MPIOLOGON.4   = 'STAT PROD;'
DD_NAME.0     = 6
DD_NAME.1     = 'MPAID'
DD_NAME.2     = 'MPRBASE'
DD_NAME.3     = 'SAMPLE'
DD_NAME.4     = 'SAMPLES'
DD_NAME.5     = 'SAMPLED'
DD_NAME.6     = 'SAMPLEE'
DA_NAME.0     = 6
DA_NAME.1     = 'MS.ASG.CMR2500.UPDATE.MPAID'
DA_NAME.2     = 'MS.ASG.CMR2500.MPRBASE.MPAID'
DA_NAME.3     = 'MS.ASG.CMR2500.SAMPLE.INDEX'
DA_NAME.4     = 'MS.ASG.CMR2500.SAMPLE.SOURCE'
DA_NAME.5     = 'MS.ASG.CMR2500.SAMPLE.DATA'
DA_NAME.6     = 'MS.ASG.CMR2500.SAMPLE.ERROR'
```

Testing your CGI Installation

ASG recommends that you perform these test steps before continuing:

Note:

If you used MPIOLOGON.0=0 in **mmr_con.cfg**, then your results may vary.

1. Access the repository from your UNIX shell.

   a. Make sure the MMR load libraries have been added to the STEPLIBENTRY of your profile file (.PROFILE) and restart your OMVS session. This is a sample definition:
   `STEPLIB=MS.MSPD.MPR.CMR2500.LOADLIB:MS.MSPDALL.REXX.LOADLIB`

   b. Verify that your settings are correct in the file **mmr_con.cfg/prf**.

   c. Execute the test file **we_shell_test.rexx** from the directory where it resides. You can use any ControlManager (CMR) command as a parameter, for example:

   ```
   /asg/cgi-bin/:
   ===>we_shell_test.rexx LIST ONLY IT-B
   ```

   You should accomplish a successful logon to the repository, and there will be a display that shows the results of your command.

If you have MMR server running, your output should look similar to Figure 2.

Figure 2 • Output with MMR-Server Running

```
** Login used :LOGON WEB PASS WEB;
***** LOGIN TO MMR-Server *****
***** YOUR MMR/ID IS : 1234567890
** Command was: LIST ONLY IT-B;
LIST OF MEMBERS
MEMBER NAME                     TYPE            USAGE CONDITION AC ALT REM OWNER
IT-BUCH-BESTELLSCHEIN          INFO-TRAEGER        3  SCE ENC   0   0   0
IT-BUCH-BESTELLUNG-KONTRLBAT   INFO-TRAEGER        3  SCE ENC   0   0   0

LIST CONTAINS       2 INFO-TRAEGER
                    2 MEMBERS IN TOTAL
  0.606507 ELAPSED TIME
```

If you do not have MMR server running, your output should look similar to Figure 3.

Figure 3 • Output without MMR-Server Running

```
*** Temporary Connection was selected ***
ASG-MMR-REXX-INTERFACE 2.3.0.2, FCT=NIKJMSPCT0001:SP TSI TLEA   T, RC=0
** Login used :
LOGON WEB PASS WEB;
DICT DEMO;
AU DEMO;
STAT DEMO;
** Command was: LIST ONLY IT-B;
LIST OF MEMBERS
MEMBER NAME                     TYPE            USAGE CONDITION AC ALT REM OWNER
IT-BUCH-BESTELLSCHEIN          INFO-TRAEGER        3  SCE ENC   0   0   0
IT-BUCH-BESTELLUNG-KONTRLBAT   INFO-TRAEGER        3  SCE ENC   0   0   0

LIST CONTAINS       2 INFO-TRAEGER
                    2 MEMBERS IN TOTAL
  0.640619 ELAPSED TIME
```

These are some reasons that your output might not be correct:

- The profile file does not contain a valid STEPLIB entry.
- The RACF settings are wrong.
- The file allocations, MMR server name, log filename, or log on settings are wrong in the file .../**mmr_con.cfg/prf**.
- The configuration filename in the file ...**we_shell_test.rexx** is wrong.

2 Access the repository from your web browser. If the first test was successful, then the REXX interface and the Web Enabler MVS components are working. This step checks the web server's configuration.

a Ensure the correct path is given in line seven of the .../**asg/html/we_cgi_test.html** file.

b Send your web browser to the URL **we_cgi_test.html**, for example: http://*your-site*/asg/webenabler/we_cgi_test.html

The URL is made up of four parts: type of protocol, IP address of the web server, valid alias of the HTML directory, and document filename.

c Enter a valid CMR command in the entry field and click Submit. The result of your command displays in the data frame.

If you have MMR server running, your output should look similar to Figure 4.

Figure 4 • Output with MMR Server Running

```
*** LOGIN *** YOUR MMR/ID IS: 1234567890;
STATUS IS DEMO NAMED 23 JUL 1999 05.04.06 UPDATE

LIST ONLY EN-A;

List contains:

                     EN-ARCHIV

                    0.838518 elapsed time
```

If you do not have MMR server running, your output should look similar to Figure 5.

Figure 5 • Output without MMR Server Running

```
CALL BPXWDYN "ALLOC FI(MPAID) DA('CUST.ASG.CMR2400.MPAID') SHR"BPXDYN RC:0
CALL BPXWDYN "ALLOC FI(MMRBASE) DA('CUST.ASG.MMRBASE.MPAID') SHR"BPXDYN RC:0
CALL BPXWDYN "ALLOC FI(MPRBASE) DA('CUST.ASG.CMR2400.MPRBASE.MPAID') SHR"BPXDYN RC:0
CALL BPXWDYN "ALLOC FI(DEMO) DA('CUST.ASG.CMR2400.DEMO.INDEX') SHR"BPXDYN RC:0
CALL BPXWDYN "ALLOC FI(DEMOS) DA('CUST.ASG.CMR2400.DEMO.SOURCE') SHR"BPXDYN RC:0
CALL BPXWDYN "ALLOC FI(DEMOD) DA('CUST.ASG.CMR2400.DEMO.DATA') SHR"BPXDYN RC:0
CALL BPXWDYN "ALLOC FI(DEMOE) DA('CUST.ASG.CMR2400.DEMO.ERROR') SHR"BPXDYN RC:0
*** Temporary connection was selected ***
STATUS IS DEMO NAMED 24 JUL 1999 05.05.06 UPDATE

LIST ONLY EN-A;

List contains:

                     EN-ARCHIV

                    2.883494 elapsed time
```

These are some reasons that your output might *not* be correct:

- The web server's start job does not contain a valid STEPLIB assignment.
- The RACF settings are wrong.
- There are invalid EXEC or PASS statements in the file .../**etc/http.conf**.
- The wrong IP address is being used.
- UNIX file permissions are not set (see step 2 on page 6).

Installing Web Enabler

The main components of Web Enabler are installed and usable.

To complete installing Web Enabler

1. Update the ACTION statements in these two files to reflect the UNIX CGI-path:
   - **s_set.html**
   - **command.html**
2. Generate your search and navigation pages. Web Enabler allows you to automatically generate HTML and some static Repository Information Model

(RIM)-specific HTML pages that are used to search and navigate. You can generate these files from the corresponding dictionary.

Your installation files contain sample files that have been constructed from the sample dictionary on the MMR installation tape. Even if you did not install the sample dictionary, the static GUI is available if you use the alias name of your UNIX HTML dictionary name as the IP address, for example:

`http://your-site/asg/`

The file index.html displays. The navigation tree is on the left side and a data frame is on the right.

The navigation tree contains hyperlinks to the sample HTML documents stored in the .../**html/sample** directory.

At this point, Web Enabler is operational with the standard defaults. Use the *ASG-Web Enabler User's Guide* to gain a clear understanding of the product's functions.

Once you understand the functions of Web Enabler, you will be prepared to customize it for your site. To enable your dictionary data, you must define your own navigation tree, adjust settings in the repository environment, and generate RIM-specific HTML forms. These subjects are discussed in Chapter 3, "Tailoring Web Enabler," on page 17.

Servlet Files

Installing and Configuring Servlet Files

The following sections provide sample procedures to install and configure Servlet with TomCat on Windows 98 or Windows NT. Alternatively, you can use the setup program delivered on CD. This setup program installs the software and the corresponding shortcuts automatically.

Note:

You must have JavaVM installed before performing these procedures.

Installing JavaVm

1. Execute `we\jvm\j2sdk_1_3_1-win.exe`.

   ASG recommends using installation folder names without special characters (e.g., blank).

2. Select the Program Files and Java Sources.

3. Finish the installation.

Installing Servlet

1. Execute `\we\wese251.exe`.
2. Complete the installation by following the displayed prompts.

Installing Tomcat

1. Select the file: `\we\tomcat3.11\jakarta-tomcat-3.1.1.tar.Z`
2. Select an installation drive to extract to the files via Winzip.

   The folders \jakarta-tomcat and subfolders are created.
3. Create 2 shortcuts which can be used to start and stop the Tomcat web server.
   - **a** Select tomcat.bat from the jakarta-tomcat\bin folder.
   - **b** Create 2 shortcuts which can be used to start and stop the server. Give the shortcuts self explaining names (e.g., Stop Server and Start Server).
   - **c** Modify the Properties of the shortcuts. Add the `run` parameter to the target entry field of the start-server shortcut and `stop` to the corresponding field of the stop-server shortcut.
4. If you are not using Windows NT, expand the environment size for DOS variables to 2048 (use the corresponding entry field of the shortcut properties dialog).

Testing the Servlet Installation

1. Start the Webserver.
2. Call start-tomcat.bat.
3. Using an Internet browser type `http://localhost:8080/examples`.

   An HTML page containing hyperlinks displays.
4. Test the samples.
5. Call stop\tomcat.bat to stop the Webserver.
6. Install the Web Enabler servlet
   - **a** Configure the Web Enabler Servlet by editing the file jakarta-tomcat\webapps\mpr\Web-inf\classes\LocalStrings.properties. Type the IP, Port, and MPR-UID/PW of the MPSF you want to communicate with.

b Configure the WebServer by editing the file jakarta-tomcat\conf\server.xml. Add an `mpr` context path. For example:

```
<Context path="/mpr" docBase="webapps/mpr" debug="0"
reloadable="true">  </Context>
```

7 Test the Web Enabler servlet using an Internet browser. Type `http://localhost:8080/mpr`.

8 Select the navigation folder and click RepositoryOnline/QueryEnv.

If there are problems with the connectivity check the following:

a The connectivity parameters the servlet will use. These are specified in the file localstrings.properties. The directory name is webapps\mpr\Web-inf\classes.

The following variables need to be set correctly. If you change them, afterwards the webserver has to be restarted.

```
asg_rep_serv.mprip_1= IP-Address
asg_rep_serv.mprpo_1_1= IP-Port
asg_rep_serv.mprpo_1_1_ID= MPR-UID
asg_rep_serv.mprpo_1_1_PW= MPR-UPW
asg_rep_serv.mprpo_1_1_Name= Descriptive Name
asg_rep_serv.mprpo_1_1_Login=dialog or hidden
asg_rep_serv.mprpo_1_1_Timeout= ReceiveTimeout in mseconds
```

(if asg_rep_serv.mprpo_1_1_Login=hidden, then no logon-dialog will be used)

b The connectivity parameters Manager Products will use:

These are specified in the exec EX-HT-HTTPCONF.

The default settings are:

```
MPHT_SERVLET = ^true^
HTML_PATH = ^/mpr/^
GIF_PATH =^/mpr/images/^
CGI_PATH=^/mpr/servlet/^
CGI_PRG=^werepserv^
HTML_INST_PATH = ^d:/jakarta-tomcat/webapps/mpr/gen/^
```

Modify the variable HTML_INST_PATH. This value depends on your installation. It must contain the folder name of your servlet installation in which static files should be deployed when you generate search dialogues.

3 Tailoring Web Enabler

This chapter provides step-by-step instructions for tailoring Web Enabler to your site's needs, and contains these sections:

Defining the Navigation Tree 18

Adjusting Your Repository Environment 19

Tailoring Dynamic HTML Generation 21
Adjusting HTML Layout 21
Multiselection 22
Search 22
Object-type-specific Settings 22

Generating RIM-specific HTML Forms 25
Using a Servlet Connection 26
Object-type-specific Search Dialogs 26
Search Dialogs 27
Object-type-independent Search Form 28
New Object Form 29
Edit Forms 29
Editing Form Display in non-MethodManager Environments 30
Editing Form Display in MethodManager Environments 31

Building Your Own Pop-up 31

Building Your Own HTML Documents 32

Building Your Own Dialog Forms 34

HTML/CSS Processing 35

Using the Import Function 35

Web Enabler Message Display 36

Defining the Navigation Tree

You define the navigation tree using Java script. The **treecfg.js** file in your HTML directory allows you to change the contents and behavior of the tree.

To define the root directory, use these statements:

`foldersTree = gFld(p1,p2)`	Generates the root folder
`p1: string`	String to display
`p2: URL`	String to be used

To define the subdirectories, use these statements:

`c_identifier = insFld(p1,gFld(p2,p3))`	Generates a new subdirectory
`p1: p_identifier`	Name of root directory
`p2: string`	String to display
`p3: URL`	String to be used

To define a hyperlink, use these statements:

`insDoc(p1,gLnk(p2,p3,p4))`	Generates a new link to a document
`p1: p_identifier`	Name of root directory
`p2: 0 or 1`	Target is frame (0) or new browser window (1)
`p3: string`	String to display
`p4: URL`	String to be used

Figure 6 • Example of a Completed Navigation Tree Definition

```
foldersTree = gFld("<i>Web Enabler, a New Interface<i>", "asg02.htm")
level_1 = insFld(foldersTree, gFld("About", ""))
     insDoc(level_1, gLnk(0,"Web Enabler", ""))
          insDoc(level_1, gLnk(1,"Allen Systems Group", "www.asg.com"))
level_1 = insFld(foldersTree, gFld("MMR Site (replicated objects) ", ""))
   level_2 = insFld(level_1, gFld("Help", ""))
        insDoc(level_2, gLnk(0, "Publications", "PUBS0000.htm"))
        insDoc(level_2, gLnk(1, "The Status Facilities", "STATUS0000.htm"))
        insDoc(level_2, gLnk(0, "The Amend Command", "AMEND000.htm"))
        insDoc(level_2, gLnk(0, "Command Syntax", "SYNCON0000.htm"))
        level_3 = insFld(level_2, gFld("Commands", ""))
```

```
insDoc(level_3, gLnk(0, "ControlManager", "SYNCOMCMR.htm"))
insDoc(level_3, gLnk(0, "DataManager", "SYNCOMDMR.htm"))
```

If you want your URL to contain a direct link to a special command, use this syntax in the **treecfg.js** file:

```
http://your-site/asg-cgi/call_mmr.rexx?OLIST+LIST+CURRENT+ONLY+EN-
```

Adjusting Your Repository Environment

You must define the specific settings for Web Enabler in the repository member EX-HT-HTTPCONF by updating it. The samples in this section contain the default settings. Although the default settings will probably run, you should amend them to fit your environment.

To amend the settings, assign the correct values to these variables in your web server environment:

Variable	Value Needed
HTML_PATH	The alias of the UNIX HTML directory.
CGI_PATH	The alias of the UNIX CGI directory.
CGI_PRG	The name of the CGI-Program (**call_mmr.rexx**)
GIF_PATH	The alias and real name of your GIF directory.
HTML_INST_PATH	The real name of the target directory static pages that will be found after generation.
HTML_GEN_FILE_EXTENTSION	The name of the file extension for generated static pages.

These are example settings:

```
HTML_PATH=^/asg/^
CGI_PATH=^/asg-cgi/^
GIF_PATH=^/asg/gif/^
HTML_INST_PATH=^/asg/html/gen/^
```

To assign default settings to your environment

1. Assign a default set of display attributes stored in the global array DISPLAY_ENTRY.

   These are some example settings:

   ```
   DISPLAY_ENTRY(1) = ^DESCRIPTION^
   DISPLAY_ENTRY(2) = ^CATALOG^
   DISPLAY_ENTRY(3) = ^NOTE^
   DISPLAY_ENTRY(4) = ^ALIAS^
   ```

   The layout of the attribute-name and its contents is influenced by the variables DEFAULT_TITLE_LAYOUT and DEFAULT_CONTENTS_LAYOUT.

   These are some sample settings:

   ```
   DEFAULT_TITLE_LAYOUT =^<B><FONT SIZE=2 COLOR=brown>^
   DEFAULT_CONTENTS_LAYOUT =^<B><FONT SIZE=2 COLOR=navy>^
   ```

   The above settings can be overwritten by object-type specific settings.

2. Assign values to the variables for default pop-up menu contents. Refer to "Building Your Own Pop-up" on page 31 for detailed explanation of these values.

   These are some example settings for a single selection stored in the global array MENU_ENTRY:

   ```
   MENU_ENTRY(1) = ^'call_mmr.rexx?DISP'>Object Report^
   MENU_ENTRY(2) = ^'call_mmr.rexx?ODISP00'>Tailored Display^
   MENU_ENTRY(3) = ^'call_mmr.rexx?ODISP02'>Tailored Print^
   MENU_ENTRY(4) = ^'call_mmr.rexx?ODISP01'>Tailored Explore^
   MENU_ENTRY(5) = ^'call_mmr.rexx?ODISP03'>Tailored View^
   MENU_ENTRY(6) = ^'call_mmr.rexx?EREFA">B_Explore^
   MENU_ENTRY(7) = ^'call_mmr.rexx?EUSA">F_Explore^
   MENU_ENTRY(8) = ^'call_mmr.rexx?HUPD">Edit^
   MENU_ENTRY(9) = ^'call_mmr.rexx?CMND+REPORT'>Report^
   ```

   These are some example settings for a multiple selection stored in the global array LIST_TYPE_ENTRY (this array is active only if the value of the variable MULTI_SELECT is TRUE):

   ```
   LIST_TYPE_ENTRY(1) = ^'CMND ALSO KEEP IN &P0 LIST MEMBERS
   '>Create list of selected^
   LIST_TYPE_ENTRY(2) = ^'CMND PRINT'>Print^
   LIST_TYPE_ENTRY(3) = ^'DISP'>Object Report^
   LIST_TYPE_ENTRY(4) = ^'ODISP00'>Tailored Display^
   LIST_TYPE_ENTRY(5) = ^'ODISP02'>Tailored Print^
   LIST_TYPE_ENTRY(6) = ^'LSTR'>Impact Analysis^
   LIST_TYPE_ENTRY(7) = ^'CMND REMOVE'>Delete^
   ```

These are some example settings for the object type MPAID-KEPT-DATA stored in the global array MKEPT_ENTRY:

```
MKEPT_ENTRY(1) = ^'call_mmr.rexx?CMND+MP+PRINT+KEPT'>Print^
MKEPT_ENTRY(2) = ^'call_mmr.rexx?CMND+FETCH+KEPT+AS+&P0+FROM'>
Create Session List
MKEPT_ENTRY(3) = ^'call_mmr.rexx?CMND+MP+REMOVE+KEPT">Remove^
```

These are some example settings for the object type SESSION-KEPT-DATA stored in the global array TMPKEPT_ENTRY:

```
TMPKEPT_ENTRY(1) = ^'call_mmr.rexx?CMND+LIST+KEPT+IN'>Print^
TMPKEPT_ENTRY(2) = ^'call_mmr.rexx?UNION+&P0'>Union with^
TMPKEPT_ENTRY(3) = ^'call_mmr.rexx?INTERSECT+&P0'>Intersect
with^
TMPKEPT_ENTRY(4) = ^'call_mmr.rexx?DIFFERENCE+&P0'>Difference
with^
TMPKEPT_ENTRY(5) = ^'call_mmr.rexx?CMND+ERASE+KEPT+LIST'>
Remove^
TMPKEPT_ENTRY(6) = ^'call_mmr.rexx?CMND+KEEP+IN+&P0+LIST+KEPT+
IN'>Copy^
TMPKEPT_ENTRY(7) = ^'call_mmr.rexx?STORE+&P0'>Store list as^
```

Tailoring Dynamic HTML Generation

Using variables, you can make these modifications to your Dynamic HTML:

- Enable multiselection
- Change the color or font of the generated HTML output
- Enable the display of search conditions with the search result
- Define object-type-specific pop-up

Adjusting HTML Layout

Use these variables to change the color or font of online generated HTML output:

Variable	Description
GLOBAL MPHT_ODISP_BGCOLOR	Sets the background color of an object report.
GLOBAL MPHT_OREF_BGCOLOR	Sets the background color of an object impact panel.
GLOBAL MPHT_CLUSTER_TEXTCOLOR	Sets the color of text in a cluster.

Variable	Description
GLOBAL MPHT_GENERAL_HEADING	Sets the heading of a data frame.
GLOBAL MPHT_OLIST_BG	Sets he background color of a search list table.
GLOBAL MPHT_OLIST_H_COLOR	Sets the font of the heading in a search list
GLOBAL MPHT_OLIST_A_COLOR	Sets the font of text in a search list. For example, `<I><B><FONT COLOR=RED size=3>`
GLOBAL MPHT_STYLE	Name of CSS file/default is wedata.css

Multiselection

To enable multiselection, set the variable MULTI_SELECT to TRUE using this command:

```
MULTI_SELECT=^true^
```

Search

To enable display of search conditions on top of a search result, set the variable SEARCH_INFO to TRUE using this command:

```
SEARCH_INFO=^true^
```

Object-type-specific Settings

Pop-ups

There are two steps to defining object-type-specific pop-ups. First, assign the object types by filling the global array POPUP with the available object types, for example:

```
GLOBAL POPUP
POPUP(1) = ^ENTITY^
POPUP(2) = ^ITEM^
POPUP(3) = ^LIST_TYPE^
POPUP(4) = ^MY_TYPE^
POPUP(5) = ^MPKEPT^
POPUP(6) = ^TMPKEPT^
```

Then define the specific pop-up contents by filling the global arrays *objecttype*_ENTRY, for example:

```
GLOBAL ITEM_ENTRY
ITEM_ENTRY(1) = ^'call_mmr.rexx?DISP'>Display^
ITEM_ENTRY(2) = ^'call_mmr.rexx?CMND+PRODUCE+COBOL+FROM'>COBOL
Structure^
ITEM_ENTRY(3) = ^'call_mmr.rexx?ODISP01'>Display V1^
ITEM_ENTRY(4) = ^'call_mmr.rexx?ODISP02'>Display V2^
ITEM_ENTRY(5) = ^'call_mmr.rexx?HUPD'>Edit^
```

Attribute Set

You can define an object type-specific set of attributes to display using *object-type*_DISPLAY_ENTRY, for example:

```
GLOBAL EXECUTIVE_ROUTINE_DISPLAY_ENTRY
EXECUTIVE_ROUTINE_DISPLAY_ENTRY(1) = ^CONTENTS^
```

Attribute Content Display

The layout of the attribute-name and its contents is influenced by the variables *object_type*_DISPLAY_ENTRY_TITLE(*n*) and *object_type*_DISPLAY_ENTRY_CONTENTS(*n*).

The value of *n* relates to the array count of the set of attributes to display.

```
EXECUTIVE_ROUTINE_DISPLAY_ENTRY_TITLE(1) = ^<I>Source Code</I>^
EXECUTIVE_ROUTINE_DISPLAY_ENTRY_CONTENTS(1) = ^<FONT FACE='Arial' SIZE=3
color=green>^
```

The above sample settings are only used by the EXECUTIVE-ROUTINE member type. If an object of that type is used by a tailorable display function of Web Enabler, the value of the attribute CONTENTS displays in green with the heading Source Code.

Redefine Member Type

You can overwrite the RIM-object-type of an object to allow specific settings such as, pop-up menus or display definitions, depending on their contents. The GUI uses either the Repository Information Model (RIM) or a corresponding entry in HTTPCONF. For example, the RIM member type ENTITY can be overwritten with the virtual member type SUBENTITY if a certain condition is true. The global REDEF_*type* must be defined; it should contain the name of an executive, for example:

```
GLOBAL REDEF_ENTITY
REDEF_ENTITY = ^MY_EXEC^
```

The executive MY_EXEC is called when an object of type ENTITY is selected for display. The name of the object is in &P0 and the global variable MPHT_MT can be set to another object-type which has separate settings in EX-HTTPCONF.

Redefine Member Display

To redefine member name display layout, use a global command of the type shown below. It must contain the name of an executive, for example:

```
REDEF_object-type_NAME
GLOBAL REDEF_INFOBANK_PANEL_NAME
REDEF_INFOBANK_PANEL_NAME = ^MY_D_EXEC^
```

MY_D_EXEC sets the global MPHT_DN. &P0 contains the object name.

To redefine member type display layout, use a global of this type:

```
REDEF_object-type_TYPE
GLOBAL REDEF_INFOBANK_PANEL_TYPE
REDEF_INFOBANK_PANEL_TYPE = ^MT_T_EXEC^
```

MT_T_EXEC sets the global and DN. &P0 contains the object type.

If set, the variables MPHT_DN and MPHT_DT display in the browser instead of member-name and member-type.

Redefine Reference Attributes

The default is to display all direct references and usages with their type. You can overwrite the default settings in HTTPCONF. The set of reference types and their display are definable using exit routines.

Define Type of Reference

You can define what type (usage or reference) of reference to process by creating a global variable that appends the string _USAGE or _REF to the name of the RIM member type. The settings relate to the referred object.

The value Y or N can be assigned, for example:

```
GLOBAL INFOBANK_PANEL_USAGE
GLOBAL INFOBANK_PANEL_REF
INFOBANK_PANEL_USAGE = ^N^
INFOBANK_PANEL_REF = ^N^
```

The above settings hide all references of an object to objects of type infobank.

To enable and manipulate the display of specific relationships, create a global variable using the member type plus _REF_ or _USAGE_ plus the relationship clause. The value of the global variable must be the name of an executive member that gets in &P0 the object name and sets in the variable MPHT_DA the string to display instead of the relationship clause, for example:

```
GLOBAL INFOBANK_PANEL_REF_SELECT
INFOBANK_PANEL_REF_SELECT = ^HT-INF-SE^
```

This example displays only the select-clause. If the executive HT-INF-SE sets the variable MPHT_DA to Choose, the user sees Choose instead of Select.

Define Level of Impact

The default level is set to 5. Use an exit to change the defaults for each member type. Specify the name of the corresponding executive in HTTPCONF using the REDEF_ plus member type plus _USA_LEVEL or _REFA_LEVEL. The executive member sets in &P0 the member name and sets the global MPHT_USA_LEVEL or MPHT_REFA_LEVEL, for example:

```
GLOBAL REDEF_ITEM_USA_LEVEL
REDEF_ITEM_USA_LEVEL=^HT-LEVEL^  /* set MPHT_USA_LEVEL  */
GLOBAL REDEF_ITEM_REFA_LEVEL
REDEF_ITEM_REFA_LEVEL=^HT-LEVEL^ /* set MPHT_REFA_LEVEL */
```

Executive IT-LEVEL:

```
MPXX LITERAL=^
MPHT_USA_LEVEL=2
MPHT_REFA_LEVEL=3
```

In the above example, the member type ITEM, the usage level is 2 and the reference level is 3.

Generating RIM-specific HTML Forms

To generate RIM-specific HTML forms

1. Create a temporary sequential file (FB, 255) named ***uid.***HTML.XFER to transfer the dialogs to the UNIX environment. The EXEC HT-ALLOC uses this filename during generation.

2. Specify the UNIX target directory in the variable HTML_INST_PATH, which is set in HTTPCONF. This is an example definition (part of the EXEC EX-HT-HTTPCONF):

   ```
   HTML_INST_PATH="/asg/html/gen"
   ```

3 Specify the UNIX file extension in the HTML_GEN_FILE_EXTENSION variable:

```
HTML_GEN_FILE_EXTENSION = ^htm^
```

Using a Servlet Connection

The Web Enabler servlet writes a file when the string '((write_to_file=$HTML_INST_PATH$filename))' is received via TCP/IP.

Therefore, the commands which generate static HTML-Files have to be issued from the Web Enabler command window of your browser. The directory specified in the variable HTM_INST_PATH will be used to store the files (see step 7 on page 15).

These are the commands you will use when creating Search or Navigation Documents:

Cluster Navigation: HT-GENCL or HT-GENCLX
Search Documents: HT-GENSA, HT-GENST or HT-GENS
Create Object: HT-GENNO

Object-type-specific Search Dialogs

A cluster selection dialog is a converted 3270 MMR cluster selection screen. If you have not defined the MMR cluster or MMR naming conventions, you can use the UNIX files named cluster.htm and s_OBJECTTYPE.htm (found in the sample directory) as examples.

Before you generate the cluster selection, the administration dictionary must be open and the executable file HTTPCONF must be called to set the global variables HTML_INST_PATH and GIF_PATH. Use this command to generate the cluster selection:

```
HT-GENCL ddname-of-hierarchy;
```

This command generates a file named cluster.htm in the UNIX environment, and a user member named HT-GENS on the MP-AID. The user member can be used to generate search dialogs for all cluster objects. If you do not want to use your MMR definitions, use this command to generate the file named cluster.htm and the user member HT-GENS on the MP-AID:

```
HT-GENCLX arg1[BUFFER];
```

where *arg1* is the name of an executive that specifies these global variables (where *n* is the counter for a cluster):

Variable	Value
MENU_TITLE_SHORT_*n*	The short term of the cluster.
MENU_TITLE_LONG_*n*	The long term of the cluster.
MENU_CONTENTS_*n*	An array containing the contents of the cluster. This variable contains two values separated by a comma; the first specifies the object type and the second specifies the prefix of the object type.

If the optional specification BUFFER is used, the output displays in a buffer only; otherwise the output is written to the UNIX file named **cluster.htm**.

This is an example executive EX-HT-GEN0XC:

```
EXECUTIVE
MPAID-NAME HT-GEN0XC
CONTENTS
MPXX LITERAL=^
GLOBAL MENU_TITLE_SHORT_1 MENU_TITLE_LONG_1 MENU_CONTENTS_1
GLOBAL MENU_TITLE_SHORT_2 MENU_TITLE_LONG_2 MENU_CONTENTS_2
MENU_TITLE_SHORT_1 = ^AD^
MENU_TITLE_LONG_1  = ^Application Development^
MENU_CONTENTS_1(1) = ^File, FI-^
MENU_CONTENTS_1(2) = ^Item, IT-^
MENU_CONTENTS_1(3) = ^Group, GR-^
MENU_TITLE_SHORT_2 = ^DM^
MENU_TITLE_LONG_2  = ^Data Modeling^
MENU_CONTENTS_2(1) = ^Entity, EN-^
MENU_CONTENTS_2(2) = ^VIEWSET, VS-^
MENU_CONTENTS_2(3) = ^BUSINESS-ITEM, BI-^
```

Search Dialogs

The file named **cluster.htm** points to several pages via hyperlink. Each entry in a cluster refers to a search dialog of an object type. The data dictionary must be open if you wish to create a search dialog. Search dialog HTML pages are generated with this command:

```
HT-GENS
```

This command creates a UNIX file for each object type and names it with the prefix s_ followed by the name of the object type and the .**htm** suffix (i.e., **s_*object-type*.htm**).

To generate a single search dialog, use this command:

```
HT-GENST arg1 arg2 arg3;
```

where:

arg1 is the RIM object type.

arg2 is the object prefix.

arg3 is the name of an executive.

EX-HT-GENSTC is a sample executive. Search fields and additional display clauses are defined in this member. This executive generates the search fields NAME, ALIAS, CATALOGUE, and DESCRIPTION. The list box Additional Display contains ALIAS, CATALOGUE, DESCRIPTION, and NOTE.

You can overwrite the default settings or create a new executive containing the two global arrays: SEARCH_FIELD and ADD_DISPLAY. The arrays should contain the corresponding attribute names. Association attributes are not allowed. The array SEARCH_FIELD can have a maximum of four entries, for example,

```
MPXX LITERAL = ^
SEARCH_FIELD(1) = ^ALIAS^
SEARCH_FIELD(2) = ^NOTE^
SEARCH_FIELD(3) = ^DESCRIPTION^

ADD_DISPLAY(1) = ^ALIAS^
ADD_DISPLAY(2) = ^NOTE^
ADD_DISPLAY(3) = ^STANDING^
ADD_DISPLAY(4) = ^AUTHOR^
```

If CATALOGUE is defined as a search field, the generator searches for valid catalogs of the corresponding object type in the dictionary, and inserts them into a catalog list box.

Object-type-independent Search Form

To generate an object-type-independent search form from a running MPR environment with the data dictionary open, use this command:

```
HT-GENSA arg1
```

where *arg1* is the name of an executive.

This command generates the file **s_PROP.htm** and places it in the UNIX environment. Association attributes are not allowed. The array SEARCH_FIELD() can have a maximum of four entries. If CATALOGUE is defined as a search field, the generator searches for valid catalogs of the corresponding object type in the dictionary, and inserts them into a catalog list box.

New Object Form

Use the HT-GENNO command to generate a create-object form. The data dictionary must be open to use this command. The UNIX file **obj_new.htm** is generated.

Edit Forms

Edit forms are dynamically generated for each task. If the MMR-assisted update functionality is available you can specify if a full or partial edit is used for each member type. In non-MethodManager environments a BLT edit is available.

There are three variables: HT-ATYPE, HT-ASIZE, and HT-AVAL. The possible values for HT-ASIZE and HT-AVAL depend on the value of HT-ATYPE, as shown in this table:

Variable	Purpose	Possible Values	
HT-ATYPE	Dialog type.	ME	Multi-line entry field.
		SE	Single line entry field.
		ML	Multi-selection list box.
		SL	Single selection list box.
		HIDDEN	Hidden field.
HT-ASIZE	Dialog property.	If ATYPE = ME, then ASIZE is size of edit area: ROWS = *x* and COLS = *y* If ATYPE = SE, then ASIZE is maximum size of field: MAXLENGTH = *x* and SIZE = *y* If ATYPE = ML, then ASIZE is lines of list entries: SIZE = *x*MULTIPLE If ATYPE = SL, then ASIZE is lines of list entries: SIZE = *x* If *x*=1, the single selection list box becomes a drop-down list box.	
HT-AVAL	Predefined string.	Usually used for list box types ML and SL. The values must be separated by blanks. During edit they will be inverted if the value is equal to the attribute content.	

This is an example definition:

```
IF ARG(1)   = #CATALOGUE# THEN DO
  HT_ATYPE = #ML#
  HT_ASIZE = #SIZE=5 MULTIPLE#
  HT_AVAL  = #MMRI MMR TSS HTML UPD EDIT#
  RETURN
END
```

Partial edit:

```
-O_ENTITY
  IF ARG(1) = #DESCRIPTION# THEN DO
   HT_ATYPE = #ME#
   HT_ASIZE = #ROWS=10 COLS=80#
   HT_AVAL  = ##
   RETURN
  END
  HT_ATYPE = #HIDDEN#
  RETURN
..
```

The above settings define the attribute DESCRIPTION of the member type ENTITY for the amend function. All other attributes of that member type are hidden.

Editing Form Display in non-MethodManager Environments

Use the Update Object form to suppress attributes. Only suppress attributes that are never used in the corresponding member types, but are defined in the RIM.

Define changes in the EXECUTIVE-ROUTINE EX-MPHTEXCLD.

The MDG_HTEXCL variable must use the UDS-Table name of the Repository for the first column, the member type name for the second column, and the attribute name for the third column.

You can insert an asterisk (*) instead of a member type name in the second column. The corresponding attribute is ignored for all member types of that Repository. For example:

```
MDG_HTEXCL(001) = ^DU777       *         UDR1^
MDG_HTEXCL(001) = ^DU777     GROUP       UDR1^
                     |         |           |
                   UDST   membertype   attribute
                             or
                          in general
```

Editing Form Display in MethodManager Environments

The System Administrator can change the entry field type for each attribute using the executable file EX-HT-ATYPE.

Building Your Own Pop-up

Each object type can have its own pop-up (as described in "Object-type-specific Settings" on page 22). To define the pop-up, change the executable EX-HTTPDCONF to use this syntax:

```
OBJECT_TYPE_ENTRY(n) = ^'call_mmr.rexx?arg1'>arg2^
```

where:

arg1 is one of these keywords

CMND	Executes your executive and uses its preformatted output.
HTML	Executes your executive and writes the unchanged output to the client.
HUPD	Generates an edit buffer of object.
HAMD	Generates an amend buffer of object.
DISP	Generates an object report (a direct impact analysis with pop-up hyperlinks and an MMR report).
ODISP00	Generates a tailored display (a pop-up of the object and indented display clauses).
ODISP01	Generates a tailored explore (display clauses and a tree of direct impacts).
ODISP02	Generates tailored print (display clauses and hyperlinks of direct impacts).
ODISP03	Generates a tailored view (display clauses only).
ODISP04	Generates a VRML view.
USA	Displays dependencies backward (table view).
EUSA	Displays dependencies backward (tree view).
REFA	Displays dependencies forward (table view).
EREFA	Displays dependencies forward (tree view).

arg2 is any string that you want displayed in the pop-up. *arg2* must be separated from *arg1* by the greater than sign (>).

Use the plus sign (+) instead of blanks if your command contains more than one word. The receiving executable file EX-CGICALL receives the command in &P0 and the object name in &P1. You can expand the keyword table for *arg1* if you extend the file EX-CGICALL with corresponding code, for example:

```
.
IF &P0 EQ ^MY_CMND^ THEN DO
       MY_EXEC &P1;
END
.
```

Note:

These keywords are for internal use and should not be used: DIALOG, DIFFERENCE, INTERSECT, KEPT, LIST, LSTR, MULTI, NEWOBJECT, OLIST, SAVE, SEARCH, STORE, and UNION.

This is the delivered definition of the default pop-up:

```
MENU_ENTRY(1)  = ^'call_mmr_rexx?DISP'>Object Report^
MENU_ENTRY(2)  = ^'call_mmr_rexx?ODISP00'>Tailored Display^
MENU_ENTRY(3)  = ^'call_mmr_rexx?ODISP01'>Tailored Explore^
MENU_ENTRY(4)  = ^'call_mmr_rexx?ODISP02'>Tailored Print^
MENU_ENTRY(5)  = ^'call_mmr_rexx?ODISP03'>Tailored View^
MENU_ENTRY(6)  = ^'call_mmr_rexx?EREFA'>B-Explore^
MENU_ENTRY(7)  = ^'call_mmr_rexx?EUSA'>F-Explore^
MENU_ENTRY(8)  = ^'call_mmr_rexx?HUPD'>Edit^
MENU_ENTRY(9)  = ^'call_mmr_rexx?CMND+MY_EXEC'>MyReport^
MENU_ENTRY(10) = ^'call_mmr_rexx?HTML+MY_H_EXEC'>My_H_Report^
```

Building Your Own HTML Documents

You can construct HTML documents with the MMR-procedures language. The HTML or Script source must be written to stdout with the SAY directive, for example:

```
MPXX LITERAL = ^
SAY ^<HTML>^
SAY ^<BODY>^
SAY ^<CENTER>^This is my first document</CENTER><BR>^
SAY ^<CENTER>^Command was ^&P0^, Selected object was ^&P1^</CENTER>^
SAY ^<BODY>^
SAY ^<HTML>^
```

There are some Web Enabler executable files available that can be adapted for use with your own generators. These are their MP-AID names (the source names have the prefix EX-):

Executable	Description
HT-OTOP *arg1*	Writes the header of an HTML document. *arg1* is the title.
HT-OBOT	Writes the footer of an HTML document.
HT-INFO	Displays a small graphic with contextual help.
HT-OTIT01 *arg1*	Writes member name with pop-up function. *arg1* is the member name.

Use these commands in your executive:

Command	Description
HT-ODISP01 *arg1*	Retrieves and displays the attributes of a member defined in HTTPCONF. *arg1* is the object name.
HT-OREF *arg1*	Table view of a direct impact analysis. *arg1* is the object name.
HT-OSTR *arg1*	Tree view of a direct impact analysis. *arg1* is the object name.
HT-OLIST *arg1*	Generates a list output including hyperlinks. *arg1* is the list command. For example: `HT-OLIST 'LIST ONLY IT-B'`
HT-TREE	Includes a Java script file. Must be called before HT-OSTR.
HT-POPUP	Includes a Java script file. Must be called before any pop up menu displays. HT-OREF, HT-OLIST, and HT-ODISP01 need this. If HT-POPUP has been included, the following hyperlink displays the menu `<A NAME="arg1" HREF="JAVASCRIPT:void(0);" onClick="startWin('arg2','arg3')">arg4</A>` where: *arg1* is any string. *arg2* is the object name. *arg3* is the object type. *arg4* string displayed for selection.
HT-OCFG07	Includes the code to interpret the exits of HTTPCONF.

Building Your Own Dialog Forms

The first field of your dialog form must contain a hidden field with the name ACTION and the value DIALOG. The method is POST and the action contains the URL of **call_mmr.rexx**.

The executable named CGICALL receives the value DIALOG in &P0. The global variable named GL-DIALOG contains the name and the values of the form fields on separate lines.

Figure 7 is an example of the syntax for a dialog form.

Figure 7 • Sample HTML for Dialog Form

```
<HTML><BODY   Background="/asg/gif/backla.jpg">
<FORM METHOD=POST ACTION="/asg-cgi/call_mmr.rexx" target="basefrm">
<TABLE><TR><TD><IMG SRC="/asg/gif/search2.gif"></TD><TD>
<H3><FONT color-darkblue>Object Info: </HE></TD></TABLE>
<input type=hidden name="ACTION" value="DIALOG">
<table>
<TR>
<td>
ObjectName:
</td>
<TD><input name="O_NAME" size=32 maxlength=32 value="*">
</TD>
<td>
<input type=submit value=" Info ">
<input type=reset value=" Reset ">
</td></tr>
</table>
</FORM>
</BODY></HTML>
```

This is an example of the syntax for the EXEC CGICALL:

```
..
IF &P0 EQ ^DIALOG^ THEN DO
       SAY ^<BR><B>P0 = ^&P0^<BR>^
       if arrayhi(^GL_DIALOG^) GT 0 then do
          DO GL_DIALOG()
                say ^<B>^GL-DIALOG(FDO(DARRAY))^</B><BR>^
          END
          DROP GLOBAL GL_DIALOG
       end
END
..
```

Your browser output looks similar to this:

```
P0 = DIALOG
ACTION is DIALOG
O_NAME is hello world
```

HTML/CSS Processing

A Cascading Style Sheet (CSS) influences the layout of the Web Enabler data frame. These HTML tags are desribed:

- Basic document settings <body> tag
- The Web Enabler information area (#Info, #Infotable, #OnlineInfo)
- The layout of the cluster selection (DIV.segment, DIV.segHeader, a.nav)

Consult your HTML documentation for a description of how to define style sheet definitions.

The CSS filename Web Enabler uses depends on the contents of the MPR-variable MPHT_CSS_FILE.

You can set this variable to different values depending on the Manager Products user ID. This can be done in the executive EX-HT-HTTPCONF.

Using the Import Function

The import screen allows you to send data to the MPAID. The contents of the data input field are written into a user member and executed when you click Upload. You can specify the member name in the corresponding field. You can use this functionality to send and execute a replacement stream of dictionary contents.

If the Add Command field has a value, this value will be sent against the repository as a command after the upload/execute is done. The usage of this function is an optional post processing.

Web Enabler Message Display

You can call the exec EX-HT-HTMLMSG to display a message on top of the Web Enabler data frame.

The first argument of this exec has to contain the message, the second a graphics filename. You can find the samples in exec EX-HT-CGICALL. The graphic displays to the right of your message. The output of the message makes use of definitions held in the exec EX-HT-HTMLCSS.

It corresponds to the static file wedata.css (see "Adjusting HTML Layout" on page 21). That means, that you have to adjust the exec EX-HT-HTMLCSS if the Web Enabler standard Cascading Style Sheet (CSS) file is not used.

4 Replicating Repository Data

This chapter provides instructions for replicating repository data, and contains these sections:

Introduction 37

Object Views 38
Tailored Print 38
Tailored Explore 39
Object Report 39
Object Impact Analysis 39
Object Information 40

Object Navigation 40
Cluster Navigation 40
List Selection. 42
Diagram Navigation 43

Introduction

You can replicate parts of your dictionary using Web Enabler's HTML generators. The generator creates HTML files in your HFS (POSIX environment) or a Lookaside Buffer containing HTML source, which you can download and send to any other platform.

The set of concerned objects must be kept in a file named KEPT-DATA for further processing. You will usually use a logical subset of your dictionary, which represents a closed network of references. The repository object references are converted to hyperlinks. If necessary, other hyperlinks are available via context-sensitive pop-up menus.

A replicated set of repository objects leads to a set of HTML files that are connected using the HREF HTML tags. If you want to use a search engine too, you need to add HTML metatags into the HTML files by adding corresponding statements into the supplied executives. All adjustments made with the executive HTTPCONV are valid.

Settings for pop-up menus are different than those for dynamic generation. There is an example that you can amend and execute before replication to activate the settings in the executive named EX-HT-HTTPSTAT.

This is a sample of the pop-up definition syntax for static generation:

```
OBJECT_TYPE_ENTRY(n) = ^'arg1'>arg2^

GLOBAL ENTITY_ENTRY
ENTITY_ENTRY(1) = ^'REPORT-'>Report^
ENTITY_ENTRY(2) = ^'USA-'>Usages^
ENTITY_ENTRY(3) = ^'REFA-'>References^
ENTITY_ENTRY(4) = ^'PRINT-'>Print^
```

The URL that calls the pop-up is composed of the contents of the array entries and the selected object name. For example:

```
URL=arg1!!objectname!!.htm
```

where *arg1* is selected in conjunction with the command name, which generates your pages.

Object Views

All generators use a named KEPT-DATA file as input and create either a buffer and HFS file or a buffer only.

Tailored Print

Tailored Print generates tailored prints of the objects held in the KEPT-DATA. This is the syntax of the executive:

```
HT-ST-001 'KEPT IN name' [BUFFER];
```

BUFFER is optional; if it is not used, UNIX files are created. The UNIX file names are ***membername*.htm**. The separator in the buffer is (beginning in column 1):

```
'*%* HTML OF membername *%*'
```

Tailored Explore

To generate tailored explore views with direct links in the navigation tree, use this executive:

```
HT-ST-002 'KEPT IN name' [BUFFER];
```

To generate tailored explore views with pop-up links in the navigation tree, use this executive:

```
HT-ST-003 'KEPT IN name' [BUFFER];
```

BUFFER is optional in both statements; if it is not used, UNIX files are created. The UNIX file names are ***membername*.htm**. The separator in the buffer is (beginning in column 1):

```
'*%* HTML OF membername *%*'
```

Object Report

To generate tailored report views, use this executive:

```
HT-ST-004 'KEPT IN name' [BUFFER];
```

BUFFER is optional; if it is not used, UNIX files are created. The UNIX file names are ***membername*.htm**. The separator in the buffer is (beginning in column 1):

```
'*%* HTML OF membername *%*'
```

Object Impact Analysis

To generate table views of indirect references, use this executive:

```
HT-ST-005 arg1 'KEPT IN name' [BUFFER];
```

To generate a tree view of indirect references with pop-up links, use this executive:

```
HT-ST-006 arg1 'KEPT IN name' [BUFFER];
```

where *arg1* is one of these impact commands: REF, REFA, US, USA.

BUFFER is optional in both statements; if it is not used, UNIX files are created. The UNIX file names are ***impact-command*!!*membername*.htm**. The separator in the buffer is (beginning in column 1):

```
'*%* HTML OF impact-command!!membername *%*'
```

Object Information

To generate unimproved HTML of a CMR command that applies to an object, use this executive:

```
HT-ST-007 CMR-command 'KEPT IN name' [BUFFER];
```

BUFFER is optional; if it is not used, UNIX files are created. The UNIX file names are ***CMR-command!!membername*.htm**. The separator in the buffer is (beginning in column 1):

```
'*%* HTML OF CMR-command!!membername *%*'
```

Object Navigation

There are two generators that can make HTML for navigation entry. The first generator makes navigation via cluster and the second enables navigation via list selection.

A third navigation entry can be implemented by using the navigation tree as described in "Defining the Navigation Tree" on page 18.

You can also use a combination of dynamic and static pages. These pages contain data which does not change often. The VRML or Diagram-view of an object is used this way and can also be used for navigation.

Cluster Navigation

This generator makes a cluster selection page. The cluster, contents, and commands to generate referred HTML are specified with an executive argument.

The commands to generate referred HTML are stored in a user member (HT-ST-030U) that you can execute after you generate the cluster. The selections can point to the lists generated by ST-031 (see "List Selection" on page 42).

The files named segments***.js**, **ball.gif**, and **objview.gif** are used by **cluster.htm**, and can be found in the HTML and GIF directories after installation.

This is the executive for a cluster selection page:

```
HT-ST-030 executive-name [BUFFER];
```

BUFFER is optional; if it is not used, UNIX files are created.

The output of the executive is either a user member (HT-ST-030U), the file **cluster.htm**, and a buffer, or a buffer only. This will be the first line in the buffer:

```
'*%* HTML OF cluster.htm *%*'
```

The configuration file sets these global variables (where *n* is the counter for the cluster):

Variable	Value
MENU_TITLE_SHORT_*n*	The short term of the cluster.
MENU_TITLE_LONG_*n*	The long term of the cluster.
MENU_CONTENTS_*n*	An array containing the contents of the cluster. This variable contains two values separated by an equals sign (=); the first specifies the name of the referred file and the second specifies the command to generate the HTML file.

This is a sample configuration file (executive HT-ST-030C):

```
EXECUTIVE
MPAID-NAME HT-ST-030C
CONTENTS
MPXX LITERAL=^
GLOBAL MENU_TITLE_SHORT_1 MENU_TITLE_LONG_1 MENU_CONTENTS_1
GLOBAL MENU_TITLE_SHORT_2 MENU_TITLE_LONG_2 MENU_CONTENTS_2
MENU_TITLE_SHORT_1 = ^1^
MENU_TITLE_LONG_1  = ^Datamodel Accounting^
MENU_CONTENTS_1(1) = ^AC-ENTITIES=HT-ST-031 CONTENTS, MPAID-NAME
EXECUTIVE-ROUTINE 'KEPT IN EN-AC^
MENU_CONTENTS_1(2) = ^AC-ITEMS=AC-IT-EXEC^
MENU_CONTENTS_1(3) = ^AC-GROUPS=AC-GR-EXEC^
MENU_TITLE_SHORT_2 = ^2^
MENU_TITLE_LONG_2  = ^Datamodel Personnel^
MENU_CONTENTS_2(1) = ^PS-ENTITIES=PS-EN-EXEC^
MENU_CONTENTS_2(2) = ^PS-ITEMS=PS-IT-EXEC^
MENU_CONTENTS_2(3) = ^PS-GROUPS=PS-GR-EXEC^
```

The user member generated contains *arg2* followed by *arg1* of the array MENU_CONTENTS_*n*.

This is a sample user member (executive HT-ST-030U):

```
HT-ST-031 CONTENTS,MPAID-NAME EXECUTIVE-ROUTINE 'KEPT IN EN-AC'
AC-ENTITIES;
AC-IT-EXEC AC-ITEMS;
AC-GR-EXEC AC-GROUPS;
PS-EN-EXEC PS-ENTITIES;
PS-IT-EXEC PS-ITEMS;
PS-GR-EXEC PS-GROUPS;
```

This sample generates the file **cluster.htm** that references AC-ENTITIES.htm, AC-ITEMS.htm, AC-GROUPS.htm, PS-ENTITIES.htm, PS-ITEMS.htm, and PS-GROUPS.htm. You can generate these referred HTML files with the commands in the user member, which are: HT-ST-031, AC-IT-EXEC, AC-GR-EXEC, PS-EN-EXEC, PS-IT-EXEC, and PS-GR-EXEC.

List Selection

You can use Executive 31 to generate a tailored object list with direct references and pop-ups. This is the syntax of the executive:

```
HT-ST-031 arg1 object-type 'KEPT IN name' [arg2]
```

where:

arg1 is a list of attributes to be displayed, separated by a comma (**,**).

name is the KEPT DATA name.

arg2 is anything or nothing. If any character is used, the output is a UNIX file; otherwise, the output is a buffer.

If a file is generated, it is named ***arg2*!!.htm**, and uses the files named **ball.gif** and **infoball.gif**. References point to file ***objectname*!!.htm**. If the output is a buffer, the first line of the buffer contains this string:

```
*%*     HTML OF LIST  list-name     *%*
```

This is an example command:

```
HT-ST-031 CONTENTS,MPAID-NAME EXECUTIVE-ROUTINE 'KEPT IN MY_E'
MY-EXECS
```

This example creates the HFS file MY_EXECS.htm that contains a list of object names held in the KEPT DATA file MY_E, and the contents of the attributes CONTENTS and MPAID-NAME for each object. The object names refer via hyperlink to files with the names ***object-name*.htm**. The display contains a sensitive **gif** that displays a pop-up if selected.

Diagram Navigation

To view an object together with a diagram, containing its direct references, use ManagerView's Repository Diagram Generation (RDG) feature. See *ASG-ManagerView User's Guide* for more details.

The corresponding VRML diagrams are generated from diagram members and stored in a separate directory of the UNIX-HFS. Therefore a VRML view of a repository object needs a generating step and a command to view the generated diagram together with textual information. The diagram itself can be used for viewing and navigation. All the nodes in the diagram are active and contain a URL which generates a display of the node-object. To view VRML diagrams the internet browser needs a VRML plug-in. These are free and are available from the Internet.

To generate the VRML files

1. Use the DIAGRAM command to create a diagram. The seed member should be the member for which a VRML View is needed. The diagram-type is defined in the EXEC EX-HT-MPMMDGHT. For example:

   ```
   DIAGRAM my_member HT
   ```

2. Convert the ManagerView diagram to VRML using the HT-GENVDG command. The parameter of this command is the name of the diagram. HT-GENVDG uses settings held in global variables of the EXEC HTTPCONF. It must be called before converting. The variables are:

VRML_INST_PATH	->	HFS-directory-name, which will be used to store the VRML files.
VRML_URL	->	Alias name of the VRML_INST_PATH
VRML_HEIGHT	->	X-Size of the embedded VRML-Diagram
VRML_WIDTH	->	Y-Size of the embedded VRML-Diagram

   For example:

   ```
   VRML_INST_PATH=^/asg/html/sample/vrml/^
   VRML_URL=^/asg/sample/vrml/^
   VRML_HEIGHT=350
   VRML_WIDTH=450
   ```

Enable selection of VRML-view

▶ Insert a new pop-up action to make the VRML View selectable. The keyword is `ODISP04`. For example:

```
MENU_ENTRY(x) = ^'call_mmr.rexx?ODISP04'>VRML View^
```

VRML creation can be tailored by these executives:

EX-HT-GENVDG

Description: converts ManagerView diagram to VRML.

Input: diagram member

Output: new diagram member (&P0_V), wrl file (&P0.wrl), and user member containing the VRML. The user member can be used as a backup of the corresponding wrl file.

EX-HT-MPMMDGHT

Description: RDG definition for WE-VRML generation.

EX-HT-MPSCNDUSRV

Description: sets the anchor of objects (URL). For example:

```
MPSCN_VRML_OBJECT_ANCHOR = CGI_PATH^^CGI_PRG^?ODISP04+^EXT_OBJ_NAME(O)
```

EX-HT-SCANNER- RULES-VRML

Description: sets exit and wrl-prototype names. For example:

```
-EXTRACT_RULES
  LOCAL E R
* OBJECT-Types:
*          PGW-type           WRL-prototyp WRL-anchor       colour
*          ..................  ............ ..............   ...........
  E(01) = ^ENTITY             EntityZyl    OBJECT           red^
  E(02) = ^GROUP              EntityBox    OBJECT           green^
  E(03) = ^ITEM               EntityCone   OBJECT           yellow^
  E(04) = ^MEMBER             EntityZyl    OBJECT           red^
  E(04) = ^FILE               Entity       -                    ^
```

EX-HT-SCANNER-VRML-EXIT-SCENE

Description: defines the background and navigation.

5 National Language Support

You must consider EBCDIC and ASCII conversions when using National Language Support (NLS) with a Web browser and OS/390.

In general the conversion of printable characters are done via SET CHARACTER-TRANSLATION commands, in the EXEC EX-HT-WEBTR.

If your Host Code-Page is not 037 you must also specify the language to use in the EXEC EX-HT-HTTPCONF.

Syntax: `MPHT_LANGUAGE = ENGLISH / GERMAN / SWEDISH / DANISH`

The EXEC EX-HT-SPECHRS contains tables of non-printable characters and can be expanded to support further languages. You can also customize the heading of impact analysis in that executive. For example:

```
EX-HT-SPECHRS
..
 if mdg_language = ^ENGLISH^  then do
  /* SPECIAL CHARACTER SETTINGS US/ENGLISH
    KAUSRZ  = X2C(^5A^)      /* EXCLAMATION POINT
    KL_AUF  = X2C(^C0^)      /* CURLY BRACKET OPEN
    KL_ZU   = X2C(^D0^)      /* CURLYBRACKET CLOSE
    EKL_AUF = X2C(^AD^)      /* SQUARE BRACKET OPEN
    EKL_ZU  = X2C(^BD^)      /* SQUARE BRACKET CLOSE
    KBACKSL = X2C(^E0^)      /* BACKSLASH
    VERTB   = X2C(^4F^)      /* Verticalbar
   /* settings for the output of the reference commands */
    mdg_header(01) = ^Dependencies^
    mdg_header(02) = ^backward^
    mdg_header(03) = ^forward^
    mdg_header(04) = ^directly^
    mdg_header(05) = ^all^
    exit
  /* END SPECIAL CHARACTER SETTINGS US/ENGLISH
 end
..
```

Use a Web Enabler utility to detect the correct hex values you need to map to if you are having problems with some special characters in your data.

Modify the **nls_test.html** file to define special characters to the system. The corresponding hex values are returned in generated tables showing you the values to map to.

1. **Conversion from Browser to UNIX(MMR).** Enter the national characters to the file **nls_test.html** and submit the form.

   Note:

   You can use the hex values of your special characters and use each of them as *arg1* in separate SET CHARACTER-TRANSLATION INPUT commands. Detect the corresponding *arg2* and *arg3* values using the ISPF-Editor and switch hex on.

2. **Conversion from UNIX(MMR) to Browser.** Search your special characters in the table, step one has created, and use each of their hex values as *arg1* in SET CHARACTER-TRANSLATION OUTPUT commands.

   Detect the corresponding *arg2* and *arg3* values using the ISPF-Editor and switch hex on.

Sample of EX-HT-WEBTR

```
<file>
EXECUTIVE-ROUTINE
MPAID-NAME HT-WEBTR
CONTENTS
MPXX LITERAL=^
* Settings for emu-codepage 1047
MPR ^NOPRINT SET CHARACTER-TRANSLATION INPUT 43 43 43^;/* s a-uml
MPR ^NOPRINT SET CHARACTER-TRANSLATION OUTPUT 43 43 43^;
MPR ^NOPRINT SET CHARACTER-TRANSLATION INPUT 63 63 63^;/* g a-uml
MPR ^NOPRINT SET CHARACTER-TRANSLATION OUTPUT 63 63 63^;
MPR ^NOPRINT SET CHARACTER-TRANSLATION INPUT DC DC DC^;/* s u-uml
MPR ^NOPRINT SET CHARACTER-TRANSLATION OUTPUT DC DC DC^;
MPR ^NOPRINT SET CHARACTER-TRANSLATION INPUT FC FC FC^;/* g u-uml
MPR ^NOPRINT SET CHARACTER-TRANSLATION OUTPUT FC FC FC^;
MPR ^NOPRINT SET CHARACTER-TRANSLATION INPUT CC CC CC^;/* s o-uml
MPR ^NOPRINT SET CHARACTER-TRANSLATION OUTPUT CC CC CC^;
MPR ^NOPRINT SET CHARACTER-TRANSLATION INPUT EC EC EC^;/* g o-uml
MPR ^NOPRINT SET CHARACTER-TRANSLATION OUTPUT EC EC EC^;
MPR ^NOPRINT SET CHARACTER-TRANSLATION INPUT BD BD BD^;/* bracket
MPR ^NOPRINT SET CHARACTER-TRANSLATION OUTPUT BD BD BD^;
MPR ^NOPRINT SET CHARACTER-TRANSLATION INPUT AD AD AD^;/* bracket
MPR ^NOPRINT SET CHARACTER-TRANSLATION OUTPUT AD AD AD^;
MPR ^NOPRINT SET CHARACTER-TRANSLATION INPUT 59 59 59^;/* ss
MPR ^NOPRINT SET CHARACTER-TRANSLATION OUTPUT 59 59 59^;
********************  Begin Web Translation (Germany) ******************
if &envt eq ^W^ then do
/* small letter   a-UML     */
MPR ^NOPRINT SET CHARACTER-TRANSLATION INPUT  EE 43 43^;
MPR ^NOPRINT SET CHARACTER-TRANSLATION OUTPUT 43 EE EE^;
/* capital letter A-UML     */
MPR ^NOPRINT SET CHARACTER-TRANSLATION INPUT  BF 63 63^;
* MPR ^NOPRINT SET CHARACTER-TRANSLATION OUTPUT 63 BF BF^;
/* small   letter o-UML     */
MPR ^NOPRINT SET CHARACTER-TRANSLATION INPUT  39 CC CC^;
MPR ^NOPRINT SET CHARACTER-TRANSLATION OUTPUT CC 39 39^;
/* capital letter O-UML     */
MPR ^NOPRINT SET CHARACTER-TRANSLATION INPUT  77 EC EC^;
MPR ^NOPRINT SET CHARACTER-TRANSLATION OUTPUT EC 77 77^;
/* small   letter u-UML     */
MPR ^NOPRINT SET CHARACTER-TRANSLATION INPUT  76 DC DC^;
MPR ^NOPRINT SET CHARACTER-TRANSLATION OUTPUT DC 76 76^;
 /* capital letter U-UML     */
MPR ^NOPRINT SET CHARACTER-TRANSLATION INPUT  63 FC FC^;
MPR ^NOPRINT SET CHARACTER-TRANSLATION OUTPUT FC 63 63^;
/* ss                       */
MPR ^NOPRINT SET CHARACTER-TRANSLATION INPUT  62 59 59^;
MPR ^NOPRINT SET CHARACTER-TRANSLATION OUTPUT 59 62 62^;
end
********************  End Web Translation (Germany) ******************
EXIT
```

This EXEC should be called from your MMR-Profile member. The variable ENVT has the value W, if accessed using the Web.

Servlet NLS

In order to have special characters display correctly, you must use the Character translation-output settings when using a servlet to connect to Manager Products. The conversion from ASCII to EBCDIC and reverse are adjusted with SET ASCII-CHARACTER-TRANSLATION directives. These are sample settings for the German language delivered with Web Enabler.

```
SET ASCII-CHARACTER-TRANSLATION C4 63^
SET ASCII-CHARACTER-TRANSLATION D6 EC^
SET ASCII-CHARACTER-TRANSLATION DC FC^
SET ASCII-CHARACTER-TRANSLATION DF 59^
SET ASCII-CHARACTER-TRANSLATION E4 43^
SET ASCII-CHARACTER-TRANSLATION F6 CC^
SET ASCII-CHARACTER-TRANSLATION FC DC^
```

A QUERY ASCII command shows the current settings.

Sample output:

CURRENT ASCII/EBCDIC TRANSLATION

```
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
    ____________________________________________________
   |                                                    |
 0 |00 40 40 40 40 40 40 40 40 40 0A 40 40 0D 40 40| 0
 1 |40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40| 1
 2 |40 5A 7F 7B 5B 6C 50 7D 4D 5D 5C 4E 6B 60 4B 61| 2
 3 |F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 7A 5E 4C 7E 6E 6F| 3
 4 |7C C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 D4 D5 D6| 4
 5 |D7 D8 D9 E2 E3 E4 E5 E6 E7 E8 E9 AD E0 BD 5F 6D| 5
 6 |79 81 82 83 84 85 86 87 88 89 91 92 93 94 95 96| 6
 7 |97 98 99 A2 A3 A4 A5 A6 A7 A8 A9 C0 6A D0 A1 40| 7
 8 |40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40| 8
 9 |40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40| 9
 A |40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40| A
 B |40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40| B
 C |40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40| C
 D |40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40| D
 E |40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40| E
 F |40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40| F
   |____________________________________________________|

     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
```

Index

A
ASCII conversions 45
ASG-Web Enabler
    installing 3
    purpose 1
    tailoring 17
attribute
    display of contents 23
    object-type specific sets 23

C
client software requirements 4
cluster navigation 40
conventions page iv

D
defining reference types 24
diagram navigation 43
dialog forms
    building 34
displaying messages 36

E
EBCDIC conversions 45
edit forms 29

H
host software requirements 3
HTML
    building documents 32
    generating forms 25
    tailoring generation 21

I
import Function 35
installing
    ASG-Web Enabler 12
    testing 9

L
level of impact
    defining 25
list selection 42
Lookaside Buffer 37

M
member name display
    redefining layout 24
member types
    redefining RIM types 23
messages
    displaying 36
multiselection 22

N
National Language Support (NLS) 45
    sample 47
navigation
    diagram 43
navigation tree
    defining 18
NLS
    converting from Browser to UNIX(MMR) 46
    converting from UNIX(MMR) to Browser 46
    servlet 48

O
object
    navigation 40
    new forms 29
Object Impact Analysis 39
Object Information 40
Object Report 39

P
pop-ups
    building 31
    object-type specific settings 22

R
reference
    defining types 24

redefining attributes 24
repository
adjusting your environment 19
replicating data 37
requirements
client 4
host 3
software 3

S

search
dialogs 27
object-type independent forms 28
object-type specific dialog 26
server connection
defining 8
servlet
NLS 48
software requirements 3
special characters 48

T

Tailored Explore 39
Tailored Print 38
tailoring ASG-Web Enabler 17

V

variables
server connection 8

W

web browsers 2