Infor WebTop for IBM® System i™ Studio

Guide to Web Conversion Tools

Important Notices

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Infor.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Infor and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company from Infor pursuant to a separate agreement ("Purpose").

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above.

Although Infor has taken due care to ensure that the material included in this publication is accurate and complete, Infor cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

Trademark Acknowledgements

All other company, product, trade or service names referenced may be registered trademarks or trademarks of their respective owners.

Publication Information

Release: Infor WebTop for IBM System i Studio 4.4

Publication date: December 13, 2011

Document code: 20111205183747

Table of Contents

Chapter 1 Installing WebTop for IBM System i Studio 4.4

1

This chapter contains the instructions for removing previous versions of WebTop Studio and installing WebTop for IBM System i Studio 4.4.

Deleting the W3TOOLs library

If you have a previous version of WebTop for IBM System i Studio installed, before you install WebTop for IBM System i Studio 4.4, back up and delete the W3TOOLS library.

Installation Procedure

Complete the steps below to install WebTop for IBM System i Studio 4.4.

Note: Your System i user profile must have *ALLOBJ authority for you to perform these steps successfully.

WARNING! If you need to reinstall this release, you must first delete the W3TOOLS library before the reinstallation.

Below are the basic steps for installing this release.

For your convenience, we are providing screen captures on a temporary basis to assist in transitioning to our new installation package.

WARNING! Before you install this release, ensure that no users are working in the Infinium applications.

1 Double-click the W3TOOLS.exe file that you downloaded to the local PC. Wait until the Introduction page shown below is displayed.

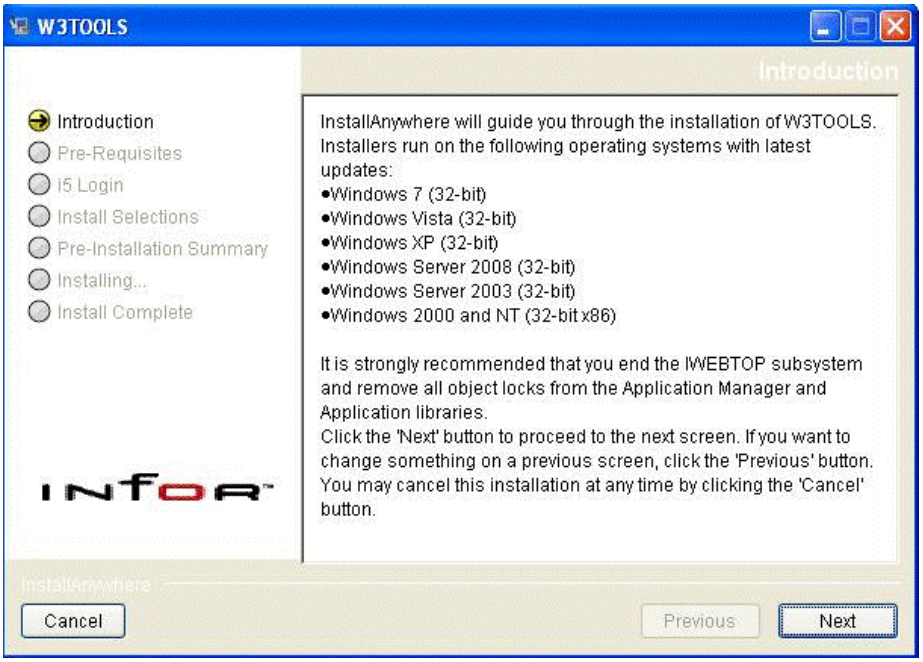

Figure 1: Introduction page

2 Review the text and click **Next** to continue. The Pre-Requisites page shown below is displayed.

Figure 2: Pre-Requisites page

The Pre-Requisites page lists the prerequisites for installing W3TOOLS.

3 Verify that you have the prerequisites listed on the page above installed on your system. If any of the prerequisite releases are not installed, cancel this installation and install the prerequisite releases first. If all prerequisite releases are installed, click **Next** to continue. The i5 Login page shown below is displayed.

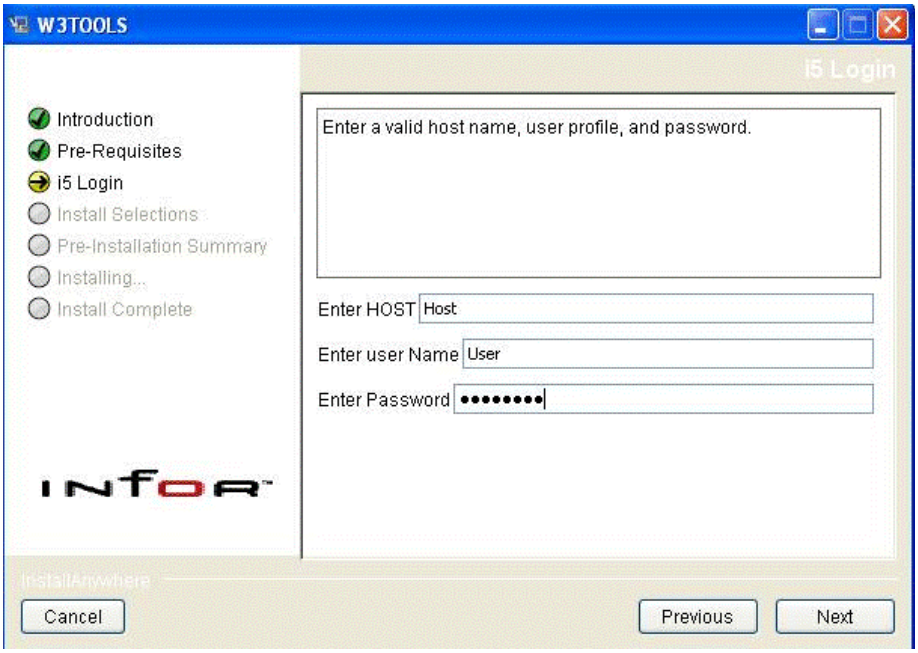

Figure 3: I5 Login page

4 Enter your system name in the **Enter HOST** field. Enter the AM2000 user ID and password in the corresponding **Enter User Name** and **Enter Password** fields.

5 Click **Next** to continue.

If the W3TOOLS library already exists, the message below is displayed. Click **OK** to cancel the installation. Rename the W3TOOLS library, and rerun the installation.

Figure 4: W3TOOLS already exists dialog

If the W3TOOLS library does not exist, the Install Selections page shown below is displayed.

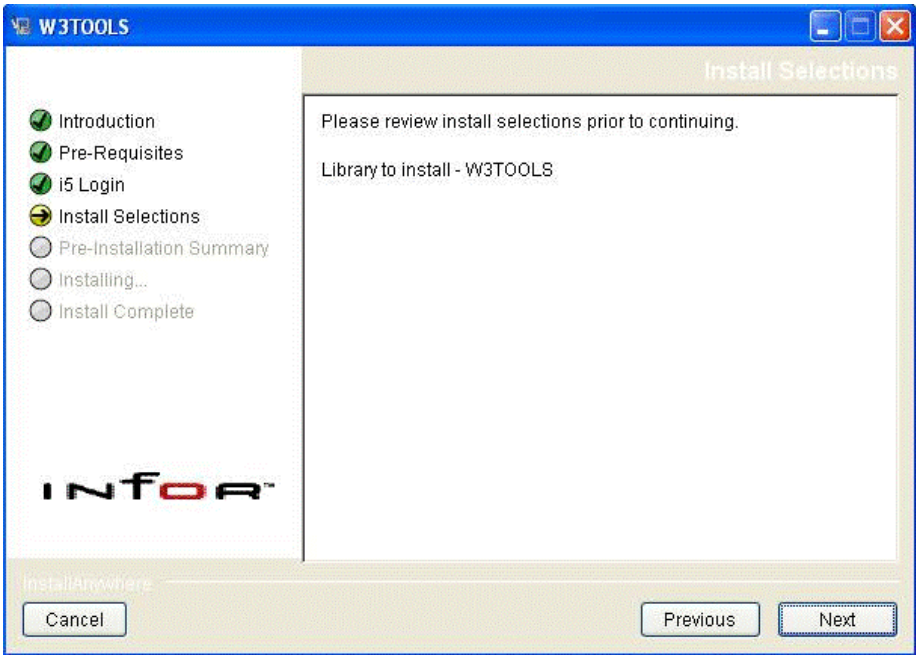

Figure 5: Install Selections page

6 Review the installation selections.

7 To make changes, click **Previous** to return to the appropriate pages and make corrections. If the selections are correct, click **Next** to continue. The Pre-Installation Summary page shown below is displayed.

Figure 6: Pre-Installation Summary page

8 Click **Install**. The Installing… page shown below is displayed.

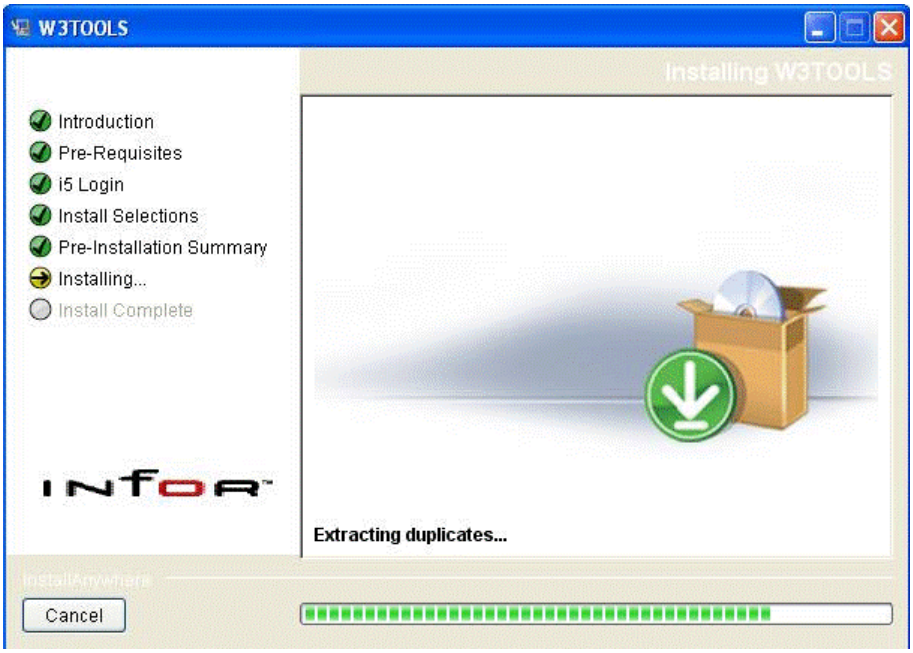

Figure 7: Installing… page

9 Wait for the Installing … page to complete. The Install Complete page shown below is displayed.

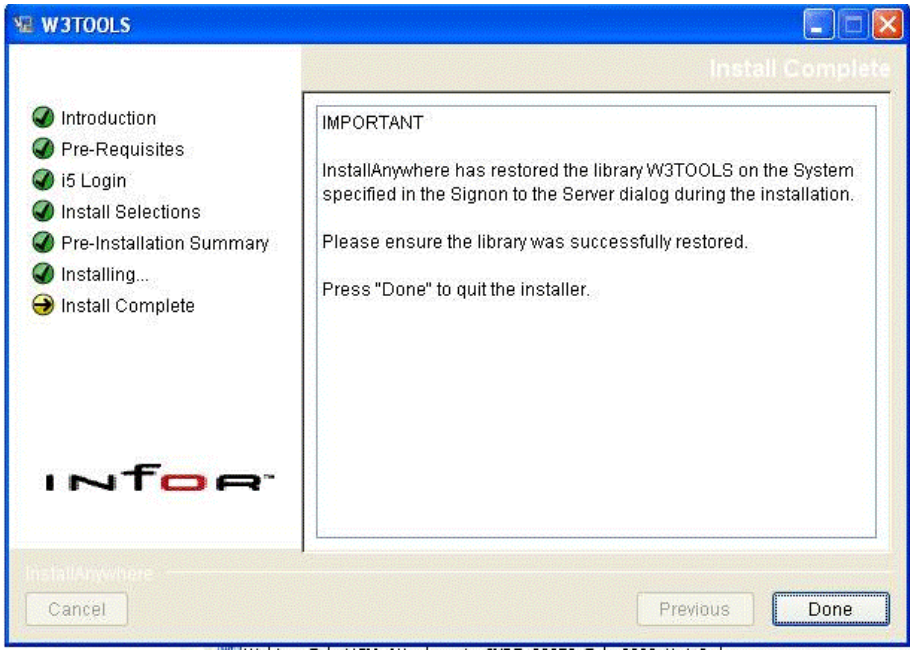

Figure 8: Install Complete page

10 Review the text and click **Done**.

11 After verifying that the installation completed successfully, delete the C:\W3TOOLS folder on your PC.

Chapter 2 Understanding the Web Architecture and Conversion Process

2

This chapter introduces the Web application architecture and the tools involved in the custom code conversion process.

The chapter consists of the following topics:

Web Architecture

The Web application environment consists of the basic high-level architecture shown below.

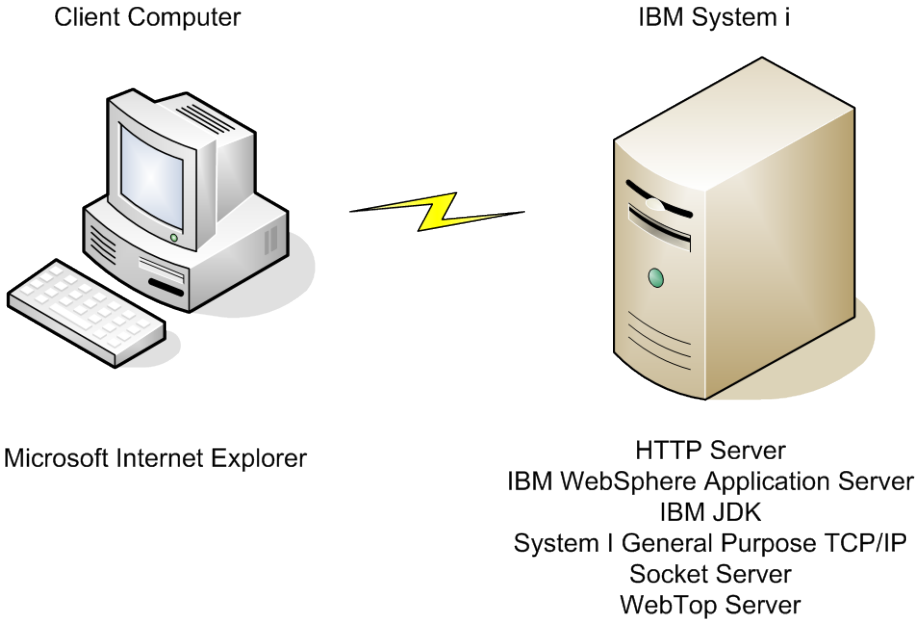

Figure 2-1: Web Architecture

Client Computers

The interface to the Web applications is through a Web browser on each client computer. No client side application installation is required.

Web/Application Server

The Web/Application Server combines the functions of a Web server and an application server. It consists of the following components:

IBM HTTP Server

The IBM HTTP Server provides a connection between the Web browser on the clients and IBM WebSphere Application Server. The Web server also is leveraged to serve static content.

IBM WebSphere Application Server

IBM WebSphere Application Server manages Java servlets, socket connections and session state controls.

System i General Purpose TCP/IP Socket Server

The System i General Purpose TCP/IP Socket Server provides the processing through which requests from WebSphere are directed to the application programs for processing.

WebTop for IBM System i Server

WebTop for IBM System i Server provides support for user profiles, security, menus, systems and versions and other application-specific controls.

ERP Applications

The ERP applications provide the processing, display and printing functions that help you to manage your business.

Tooling

Tooling consists of the processes of taking RPG IV and CL source files, adding code to them and creating the run-time data.

Conversion Process

The process for converting your custom RPG IV and CL programs for use with the WebTop for IBM System i Server consists of a set of tools. Below is a description of the tool involved in this process.

Business Tier Tool

The business tier tool converts your custom RPG IV and CL programs by inserting code that is required to display the program data within a browser. This tool does not change the logic flow within your code, it inserts code that bypasses the display file I/O and executes WebTop Server code.

You execute the business tier tool from a System i command line on your IBM System i. You must enter a set of parameters that are specific to the application you are converting.

Chapter 3 Converting Custom Programs

3

This chapter explains the business tier conversion tool that converts RPG IV and CL program code for use within a browser.

The chapter consists of the following topics:

Overview of the Business Tier Tool

Purpose

The business tier conversion tool converts existing interactive RPG IV and CL programs to enable the programs to display in a Web browser. The tool accomplishes this by circumventing the System i display file I/O by inserting code that processes and passes the application data to or from the browser through the WebTop Server Display Object Handler (DOH), System i TCP/IP Sockets Server and WebTop Server Service Object (SO).

This tool is applicable only to interactive programs that use display files.

The code that the conversion tool inserts into a program captures the data that is normally passed to the display files and sends it to the Display Object Handler, which processes and passes the data to the browser via the WebTop Service Object or Java Controller running in WebSphere Applciation Server JVM. The Display Object Handler replaces the System i Workstation Manager for data that is to be displayed in a Web browser.

After you run the conversion tool, the System i programs are set to run both in a Web browser and with 5250 emulation. The Establish Runtime Environment program, W3GERE, in the AM2000 library determines whether you are running the programs from a browser or an emulator.

The conversion tool makes no distinction between ERP application code and custom code; therefore, the tool processes both equally well. The tool uses the Infor program patterns to tool programs. The closer the customized programs are to the Infor pattern, the better the tool is able to convert and process the customized program.

Code Changes

Programs

The conversion tool inserts the following code into each converted ERP application program:

- Data elements into the Informational Data Structure for manipulating the data that would have been returned from the display screen.
- New subroutines (DO_READ and DO_WRITE) that process READ and WRITE operations to or from a screen format. This code bypasses the original read or write code.
- Code that assembles the buffer that would have gone to the display screen but which now goes to the browser. The tool constructs two routines: one for reading and one for writing. It assembles and disassembles the buffer as if it is going to the display screen. This is done within the original programs because the programs have knowledge of all of the fields. Performing this processing within the original programs also reduces the effect on performance.

These changes involve substring operations to build the buffer based on the fields that would normally have been sent to the display screen. The substring operations are in a CASE statement that is based on the format that the program is processing. At the end of the routine, the program calls the Display Object Handler.

READ operations are the opposite of WRITE operations. In WRITE operations the Display Object assembles the buffer data and in READ operations the Display Object disassembles the buffer data.

Each program receives the same code changes except for the display file-specific subroutines at the end of each program. The conversion tool makes these changes in terms of the fields used but performs the same type of processing in each program.

The conversion tool constructs the code changes. The processing code can optionally reside in Include files (copy books) outside of the main program source so that it has a minimal effect on future changes to the original source code.

If you specify to create special widgets during the Weblicate process, the business tier tool adds the decorator definitions to the WebTop Server decorator file, W3PDEC.

Refer to the *WebTop Server System Administrators Guide* for information about the types of decorators allowed and examples of the code.

Effect on the Original Programs

If you are running an application with 5250 emulation, the conversion should have no effect on the program processing since it uses the original code.

When the conversion tool runs, it inserts its code in the appropriate locations. This new code is processed only if you are running the application within a Web browser.

The tool has no effect on WebTop Server soft coded actions, NLE, CUA/SAA, formatting data, data validation, processing subfiles or any other processing.

Subfiles

For programs that use subfiles, the conversion tool enters the appropriate program code for tracking the relative record number for each subfile record. This number accompanies each record to and from the server. For Chain (CHAIN) and Read/Change (READC) operations, the program passes the relative record number of the record that it is to retrieve and then tests the results to ensure that the record was retrieved.

The display object caches the subfile data in a user index. It sends out the data and retrieves it based on the relative record number.

The display object keeps track of the last relative record number to handle scrolling forward and backward. It also keeps track of the highest record number to detect the end of the file. When you use the display object, all of this processing remains outside of the original programs.

If you use using built-in functions with display file I/O, always include the format name for the tool to handle the processing properly.

Effects on Program Size and Performance

The conversion tool adds lines of code to the original programs; therefore, the program sizes increase.

Since the new lines of code are compiled into the original programs and the primary processing performed by the conversion code are string manipulation operations, the effect on performance is minimal.

Compiling Line Number Sequence and Modification Dates

When you run the conversion tool, the tool resequences the source member line numbers.

The tool preserves the modification date for all original program code lines and assigns the conversion run date to all lines of code that the tool inserts.

Tool limitations

Refer to Appendix B for details on WebTop tool limitations.

Function Keys

The business tier tool performs the following function key procedures:

- Extracts the function key information
- Writes the information by using the API to the appropriate function keys file

Reports

The business tier tool generates the following reports that can help you analyze the conversion process:

Report	Description
W3TXR4P1	Audit report for the RPG IV programs that are converted
W3TXR4P2	Error report for the RPG IV programs that are converted
W3TXCLP1	Audit report for the CL programs that are converted
W3TXCLP1	Error report for the CL programs that are converted
W3TEXPTP1	Report listing all decorators that were generated by the weblicate command

Rerunning the Business Tier Tool

Each time you run the business tier tool the program removes the RPG IV and CL code it inserted during the previous conversion and then inserts the new RPG IV or CL code.

Running the Business Tier Conversion Tool

You run the business tier conversion tool as a System i command with the appropriate parameters specified. Online help is available.

Complete the steps below to run the business tier tool.

1. Back up your custom program library.
2. Confirm that no one is using the application you are converting.
3. After an application library list has been established, type the following:

   ADDLIBLE W3TOOLS *LAST

4. Type the System i Display Library List command, **DSPLIBL**, to display your library list. Confirm that your custom program library and W3TOOLS are in the list.
5. On a System i command line, type **WEBLICATE** and press F4.

   The following steps prompt the command to run interactively. If you want to run the job in batch, prompt within the Submit job **SBMJOB** command.

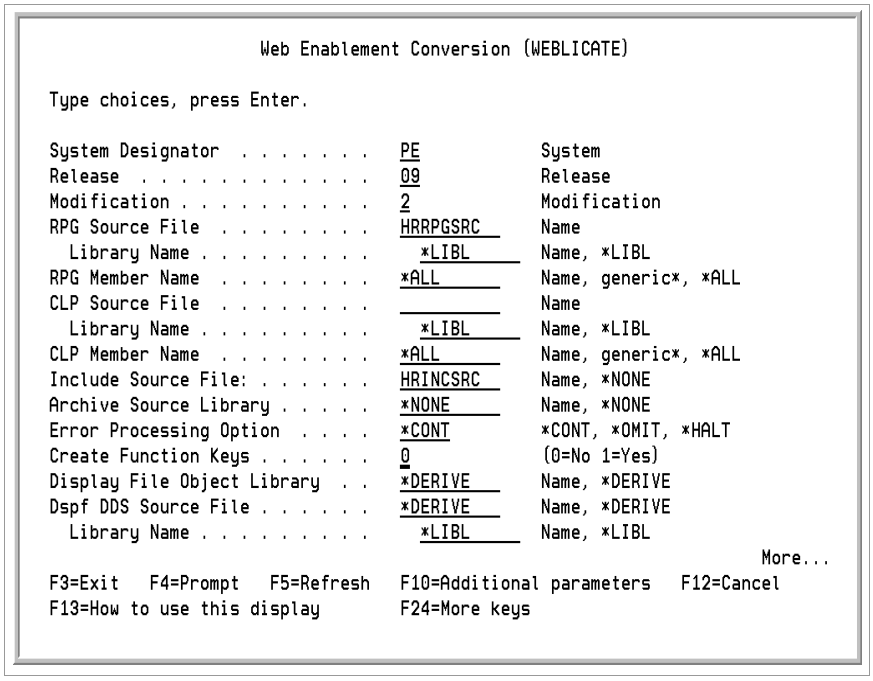

Figure 3-1: Web Enablement Conversion screen

6 Enter values for the following parameters:

System Designator

Type the two-character designator that identifies the application system you are converting. For example, type **PE** for Infinium Human Resources or **GL** for Infinium General Ledger.

Release

Type the two-digit product release number.

Modification

Type the one-digit modification number for the product release.

RPG Source File

Type the name of the source file that contains your custom RPG IV application source code.

Library Name (for RPG Source File)

Type the name of the library where the custom RPG program source file resides.

RPG Member Name

To convert all RPG source members, retain the default value ***ALL**. To convert a single source member, type the name of the member. To convert a subset of members, type a common portion of the member name followed by an asterisk (*). For example, type **PY*** to convert all members that begin with PY.

CLP Source File

If applicable, type the name of the source file that contains your custom CL application source code. Otherwise, leave blank.

Library Name (for CLP Source File)

If applicable, type the name of the library where the custom CL program source file resides.

CLP Member Name

To convert all CL source members, retain the default value ***ALL.** To convert a single source member, type the name of the member. To convert a subset of members, type a common portion of the member name followed by an asterisk (*). For example, type **PY*** to convert all members that begin with PY.

Include Source File

Type the name of the file that will contain your inserted copybook code. ***NONE** inserts copybook code into the existing program.

This file will be created if it does not exist.

Archive Source Library

If you specify an archive library, the business tier tool will archive your current source code prior to running the conversion. Type a library name unless you choose not to back up your source code. We strongly recommend that you specify an archive library. This library will be created if it does not exist.

The naming convention for the archive files that will reside within the archive library consists of the following parts:

- Positions 1–2: AR (for archive)
- Positions 3–6: MMDD (system month and day)
- Position 7: Archive type (R-RPG, P-copy book, or C-CL)

- Positions 8–9: Sequence number (is used if you already have an archive for this date)

Error Processing Option

Type the value that specifies the type of error processing you prefer. Valid values are:

***CONT** If an error is a non-terminal error, continue processing. This is the default.

***OMIT** If an error is a non-terminal error, do not convert the member.

***HALT** Cease processing when any error is encountered.

All errors are listed in the generated report.

Create Function Keys

You should create the function keys the first time you run the conversion and any time you make function key changes.

This process does not apply to soft coded applications.

Display File Object Library

Type the name of the library where your display files reside. You can use *DERIVE to determine the library automatically. A terminal error is generated if *DERIVE cannot determine the library.

The business tier tool uses this library name to create the IFS folder for the Java classes that it generates. The naming convention for this folder is as follows:

- Library name
- System
- Release
- Modification

For example, HR2000PE092.

7 Press Page Down to display additional parameters.

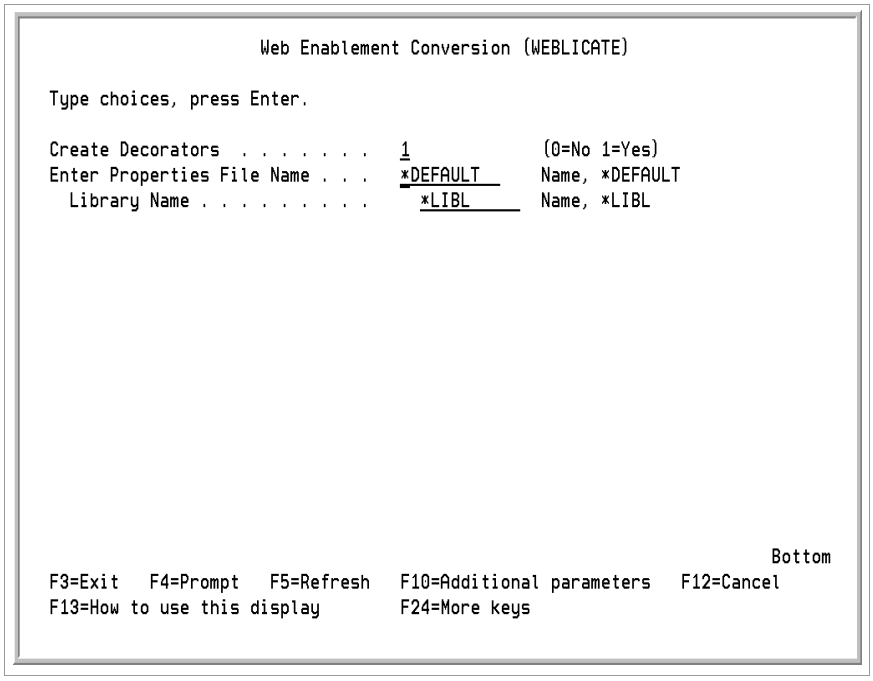

Figure 3-2: Web Enablement Conversion screen - additional parameters

Dspf DDS Source File

Type the name of the display file Data Description Specifications source file. You can use *DERIVE to determine the file. A terminal error is generated if *DERIVE cannot determine the file.

Library Name

Type the name of the display file DDS source library. You can use ***LIBL** to instruct the business tier tool to determine the library name if the library is in the library list. A terminal error is generated if the business tier tool cannot determine the library.

8 Press F10 to display additional parameters.

Create Decorators

Specify yes to insert the decorators that can be tooled from the Data Description Specifications file, DDS. Otherwise, specify no.

Enter Properties File Name

Specify the name of the properties file to use during tooling.

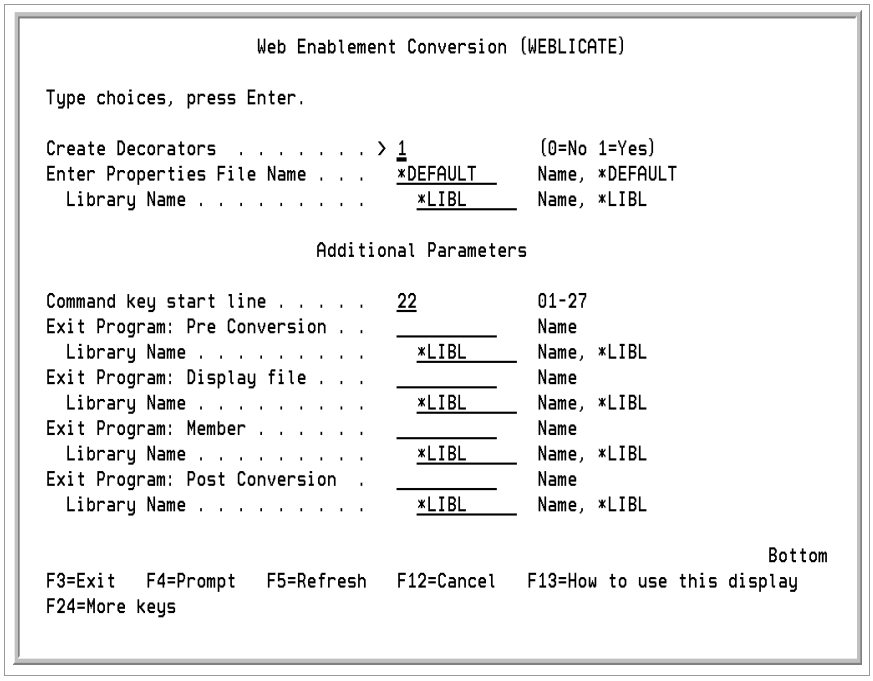

Figure 3-3: Web Enablement Conversion screen - exit programs

Command key start line

Type the starting line number where fields over 78 positions in length are considered function keys. The default is **22**.

9 To provide custom conversion processing, you can specify exit programs that run at specific times during the conversion process. The following options are available:

Exit Point	Description	Parameters
Pre-conversion	The exit program is invoked before the conversion process begins.	None
Display file	The exit program is invoked for each display file that is converted. Type **W3GHRCLNUP** if applicable. Refer to the “Function Keys” topic on Page 3-5.	System (2) Release (2) Modification (1) Display File (10)

Exit Point	Description	Parameters
Member	The exit program is invoked for each member that is converted.	System (2) Release (2) Modification (1) Member (10)
Post-conversion	The exit program is invoked after the conversion process completes.	None

Type program names and libraries for any exit programs that are applicable.

10 After you enter all applicable parameter values, press Enter to start the conversion.

11 When the conversion completes, review the conversion reports W3TXR4P1, W3TXR4P2, W3TXCLP1, W3TXCLP2 and W3TEXTP1.

12 Recompile your entire converted custom program library to incorporate the changes.

Tooled CL programs generate members in the source file QINFSNDRCV to handle the communication between the application program and the display object program W3GDOH. You must compile these members as the process does not execute a compile.

Note: The business tier process does not recompile your programs.

Appendix A Troubleshooting

A

This appendix consists of tips for resolving the following:

- Errors where the message, “Page not found” is displayed
- Errors in which no data is displayed on the page

Tips for Resolving Issues

When you are diagnosing a problem with the Web, you must identify who is experiencing the problem and that person's role. Typical issues may be associated with the following:

- Browser determines what controls can be accepted or loaded
- HTTP serves simple requests:
  - HTML pages
  - Images
  - Script and so on
- WebSphere handles the Servlets and delivers JSP
- System i Socket Services receives the TCP request and passes it to a run job which does the following:
  - Maps the request to a message
  - Calls the menu extract
  - Calls the authorized systems extract
  - Calls the job to submit an ERP Application job

Page Not Found Errors

If the user types a URL and a **Page not found 404** error is displayed, do the following:

1 Verify that the following URL is correct:

http://***hostname***:***port***/inabler/web/Login

Replace ***hostname*** with the name of your System i.

Replace ***port*** with the port number on which your HTTP Server is listening on. The default is **8010**.

2 Access the System i and verify that the HTTP configuration has started. For example, try to access the following image file:

http://***hostname***:***port***/inabler/web/httproot/images/back.gif

The system should display the back.gif image.

3 Verify that the HTTP instance is running in the QHTTPSVR subsystem. If this instance is not running, you can start it by using one of the following commands:

- **STRTCPSVR SERVER(*HTTP) HTTPSVR(*WEBTOPHTTP*)**

  Replace ***WEBTOPHTTP*** with the name of your HTTP Server.

- **HTTP://hostname:2001**

  Replace ***hostname*** with the name of your System i.

4 If you are certain that the URL is correct, the HTTP instance is running and a page is still not displayed, access the administration page by using the following URL:

http://***hostname***:***port***/inabler/Admin

The URL is case sensitive for Java. If a page is not displayed, verify that WebSphere is started.

No Data on the Page

If the user specifies the correct URL and receives a WebTop Server Menu page with no data, the issue is one of the following:

- Socket services
- WebSphere

Checking the Socket Service

The socket services run in the IWEBTOP subsystem. You must ensure the following:

- Programs TCPPGIVE, TCPPTAKE, W3GJOB and W3CCTL1 are owned by a profile that has *allobj authority. These programs must adopt authority from the user profile value, ***owner**.
- The socket server is started.
- The socket server is started on the correct port. This port is different from the HTTP port. The socket server port is defined in the /IWEBTOP.ear/IWEBTOP.war/properties/default.properties file. The directive for this is **host.port=**.

If the socket server is running and you are confident that it is accessing the correct port, try the URL again. If you still do not get data, verify that WebSphere is running.

Appendix B WebTop Studio Limitations

B

This appendix describes the WebTop Studio tool limitations.

Authorization and authentication

Authorization and authentication is based on the System i local User Registry only.

IBM WebSphere Application Server

The process will support only the IBM WebSphere Application Server at runtime. No other middleware (JVM) is supported.

COBOL

Cobol programs are not supported.

Interactive commands

OS/400 interactive system commands are not generally supported. A small subset of commonly used commands is provided for WRKSPLF, WRKMSG, WRKUSRJOB. A 5250 emulator session (green screen) must be used for these functions, including field level security administration, and F1 help and associated prompt programs available only via F1 help. No access is provided to the command line.

Status messages

Status messages are displayed at the end of an operation and not when they are occurring.

Group jobs

Group jobs are not supported.

CL constraints

CL programs can be made to call RPG, which in turn can be managed by the system and method.

RPG constraints

The process applies only to interactive RPG IV programs running on the i. CL and PLI programs can be made to call RPG, which in turn would be able to call WebTop.

RPG indicators

Record-identifying indicators are not supported.

Display file reads

Reads at the display file level are not supported.

Free format RPG IV

The free format patterns as described in the “Supported Free Form Patterns” appendix C are supported.

Source line length

Only an RPG source file line length of 112 is supported.

Source code availability

The display file source and object must be available to the conversion tool when running. The tool operations are based on the content of the display files.

Display file constraints

Format naming conventions

XML entity references are not supported in format names, for example: '@', '&', ‘’’, ‘””, ‘<’, ‘>’.

Fake subfile screens

Decorator options on fake subfile screens, where there are more than one size of field, are not supported.

Option Indicators

**Multiple indicators on option decorators are supported by using Manual Decorators. At least one non-conditioned option indicator must be present for option decorators to be inserted.

Menu keywords

Menu DDS keywords are not supported.

Blink attribute

The blink attribute is not supported.

Color attributes

The color designated as the default by attribute settings in a 5250 environment is not supported.

Field overlays

Overlaying fields from one display file to another display file is not supported.

SFLRCDNBR(*TOP)

The SFLRCDNBR with the *TOP parm is not supported

Miscellaneous keywords not supported

- DUP
- HOME
- all Menu Keywords
- all Help Keywords
- all Window function Keywords

Display files per Interactive program

The conversion tool supports up to ten display files per interactive program.

Display file format and field limits

The maximum number of formats is 100.

The maximum number of fields on a format is 256.

Notes

Appendix C Supported Free Form Patterns

C

This appendix provides details of the supported free form patterns.

The appendix consists of the following topics:

Patterns changed to support free-form RPG used for workstation I/O

This section contains tree form patterns that were changed to support free form RPG used for workstain I/O.

Assumptions and restraints

The inserted Web code if statements will start at two positions to the left of the original source start position if there is space. If not. The statements will start at position 40.

All display I/O code must be on a single line, ending with a semicolon.

When the filename is not given, the program uses the last file I/O to determine if this %found is linked to a display file. If so, the program inserts the Web code.

Write

Original code

```
write dsp01;
```

WebTop code

```
if jobtype <> 'W';
  write dsp01;
// ---- Display File I/O Redirected ---
else;
  O@ACTION = 'WRITE     ';
  o@display = 'MDDCFG     ';
  x@format = 'DSP01     ';
  exsr DO_WRITE;
endif;
//----- End -----
```

READ

Original code

```
read dsp01;
```

WebTop code

```
if jobtype <> 'W';
  read dsp01;
// ---- Display File I/O Redirected ---
else;
  O@ACTION = 'READ      ';
  o@display = 'MDDCFG    ';
  x@format = 'DSP01     ';
  exsr DO_READ;
endif;
//----- End -----
```

EXFMT

Original code

```
exfmt dsp01;
```

WebTop code

```
if jobtype <> 'W';
  exfmt dsp01;
// ---- Display File I/O Redirected ---
else;
  O@ACTION = 'WRITE     ';
  o@display = 'MDDCFG    ';
  x@format = 'DSP01     ';
  exsr DO_WRITE;
  O@ACTION = 'READ      ';
  exsr DO_READ;
endif;
//----- End -----
```

CHAIN

Original code

```
CHAIN SF2RRN SFL02;
```

WebTop code

```
if jobtype <> 'W';
  CHAIN SF2RRN SFL02;
// ---- Display File I/O Redirected ---
else;
  O@ACTION = 'CHAIN     ';
  o@display = 'MDDCFG    ';
  x@format = 'SFL02     ';
  O@SFILE = SF2RRN       ;
  exsr DO_READ;
if o@sfile = *HIVAL;
else;
  SF2RRN      = O@SFILE;
endif;
endif;
```

READC

Original code

```
readc sfl02;
```

WebTop code

```
if jobtype <> 'W';
  readc sfl02;
// ---- Display File I/O Redirected ---
else;
  O@ACTION = 'READC     ';
  o@display = 'MDDCFG    ';
  x@format = 'SFL02     ';
  O@SFILE = *ZERO;
 exsr DO_READ;
if o@sfile = *HIVAL;
else;
  SF2RRN      = O@SFILE;
endif;
endif;
//----- End -----
```

%FOUND(filename)

Original code

```
IF %FOUND(MDDCFG);
 ...
 ...
Endif;
```

Changed code

```
IF (%FOUND(MDDCFG)
   )
   ;
 ...
 ...
endif;
```

WebTop code

```
  IF (%FOUND(MDDCFG)
     )
              and jobtype <> 'W'
              or (jobtype = 'W' and
O@SFILE<>*hival)
    ;
endif;
```

%EOF(filename)

Original code

```
IF %EOF(MDDCFG);
 ...
 ...
endif;
```

Changed code

```
IF (%EOF(MDDCFG)
    )
    ;
 ...
 ...
endif;
```

WebTop code

```
IF (%EOF(MDDCFG)
   )
            and jobtype <> 'W'
            or (jobtype = 'W' and
O@SFILE<>*hival)
   ;
endif;
```

NOT %FOUND(filename)

Original code

```
IF not %FOUND(MDDCFG);
  ...
  ...
endif;
```

Changed code

```
IF (not %FOUND(MDDCFG);
  )
  ;
  ...
  ...
endif;
```

WebTop code

```
IF (NOT %FOUND(MDDCFG)
   )
            and jobtype <> 'W'
            or (jobtype = 'W' and
O@SFILE=*hival)
   ;
endif;
```

NOT %EOF(filename)

Original code

```
IF NOT %EOF(MDDCFG);
 ...
 ...
endif;
```

Changed code

```
IF NOT %EOF(MDDCFG)
   )
   ;
endif;
```

WebTop code

```
IF (NOT %EOF(MDDCFG)
   )
            and jobtype <> 'W'
            or (jobtype = 'W' and
O@SFILE=*hival)
   ;
endif;
```

%FOUND and %FOUND()

Note: When the filename is not given, the program uses the last file I/O to determine if this %found is linked to a display file, and if so, inserts the Web code.

Original code

```
IF %FOUND;
  ...
  ...
endif;
```

Changed code

```
IF (%FOUND
   )
   ;
  ...
  ...
endif;
```

WebTop code

```
IF (%FOUND
   )
                and jobtype <> 'W'
                or (jobtype = 'W' and
O@SFILE<>*hival)
   ;
  ...
  ...
endif;
```

%EOF and %EOF()

Note: When the filename is not given, the program uses the last file I/O to determine if this %found is linked to a display file, and if so, inserts the Web code.

Original code

```
IF %EOF;
...
...
endif;
```

Changed code

```
IF (%EOF
   )
   ;
  ...
  ...
endif;
```

WebTop code

```
IF (%EOF
   )
                and jobtype <> 'W'
                or (jobtype = 'W' and
O@SFILE<>*hival)
   ;
  ...
  ...
endif;
```

NOT %FOUND and NOT %FOUND()

Note: When the filename is not given, the program uses the last file I/O to determine if this %found is linked to a display file, and if so, inserts the Web code.

Original code

```
IF NOT %FOUND;
  ...
  ...
endif;
```

Changed code

```
IF (NOT %FOUND
   )
   ;
  ...
  ...
endif;
```

WebTop code

```
IF (NOT %FOUND
   )
            and jobtype <> 'W'
            or (jobtype = 'W' and
O@SFILE=*hival)
   ;
  ...
  ...
endif;
```

NOT %EOF

Note: When the filename is not given, the program uses the last file I/O to determine if this %found is linked to a display file, and if so, inserts the Web code.

Original code

```
IF NOT %EOF;
  ...
  ...
endif;
```

Changed code

```
IF (NOT %EOF
   )
   ;
  ...
  ...
endif;
```

WebTop code

```
IF (NOT %EOF
   )
            and jobtype <> 'W'
            or (jobtype = 'W' and
O@SFILE=*hival)
   ;
  ...
  ...
endif;
```

OPEN

Original code

```
open mddcfg;
```

WebTop code

```
if jobtype <> 'W';
  open mddcfg;
endif;
```

CLOSE

Original code

```
close mddcfg;
```

WebTop code

```
if jobtype <> 'W';
  close mddcfg;
endif;
```

Previously supported fixed-form patterns

This section contains the fixed form patterns that were previously supported.

Write

Original code

```
           C                   write    DSP01
```

WebTop code

```
         C     JOBTYPE       IFNE      'W'
         C                   write     DSP01
         /*  --------------------------------  Display File I/O Redirected  ----
         C                   ELSE
         C                   MOVE      'WRITE     '
         O@ACTION
         C                   MOVEL     'MDDCFG    '
         O@DISPLAY
         C                   MOVE      'DSP01     '
         X@FORMAT
         C                   EXSR      DO WRITE
         C                   END
         \*  ------------------------------------------------------  End  ----
```

READ

Original code

```
C                   read      DSP01
```

WebTop code

```
C     JOBTYPE       IFNE      'W'
C                   read      DSP01
/*  ---------------------------------  Display File I/O Redirected  ----
C                   ELSE
C                   MOVE      'READ '
O@ACTION
C                   MOVEL     'MDDCFG '
O@DISPLAY
C                   MOVE      'DSP01 '
X@FORMAT
C                   EXSR      DO READ
C                   END
\*  ------------------------------------------------------  End  ----
```

EXFMT

Original code

```
c exfmt dsp01
```

WebTop code

```
C     JOBTYPE       IFNE      'W'
c                   exfmt     dsp01
/*  ---------------------------------  Display File I/O Redirected  ----
C                   ELSE
C                   MOVE      'WRITE     '
O@ACTION
C                   MOVEL     'MDDCFG    '
O@DISPLAY
C                   MOVE      'DSP01     '
X@FORMAT
C                   EXSR      DO WRITE
C                   MOVE      'READ      '
O@ACTION
C                   EXSR      DO READ
C                   END
\*  ------------------------------------------------------  End  ----
```

CHAIN

Original code

```
c     sf2rrn        CHAIN     SFL02
```

WebTop code

```
C     JOBTYPE       IFNE      'W'
c     sf2rrn        CHAIN     SFL02
/*  --------------------------------  Display File I/O Redirected  ----
C                   ELSE
C                   MOVE      'CHAIN '
O@ACTION
C                   MOVEL     'MDDCFG    '
O@DISPLAY
C                   MOVE      'SFL02     '
X@FORMAT
C                   EVAL      O@SFILE     =
SF2RRN
C                   EXSR      DO READ
C     O@SFILE       IFEQ      *HIVAL
C                   ELSE
C                   EVAL      SF2RRN      =
O@SFILE
C                   END
C                   END
\*  ------------------------------------------------------  End  ----
```

READC

Original code

```
c                   readc     sfl02
99
```

WebTop code

```
C     JOBTYPE       IFNE      'W'
c                   readc     sfl02
99
/*  --------------------------------  Display File I/O Redirected  ----
C                   ELSE
C                   MOVE      'READC     '
O@ACTION
C                   MOVEL     'MDDCFG    '
O@DISPLAY
C                   MOVE      'SFL02     '
X@FORMAT
C                   EVAL      O@SFILE     =
*ZERO
C                   EXSR      DO_READ
C     O@SFILE       IFEQ      *HIVAL
C                   MOVE     '1'
*IN99
C                   ELSE
C                   MOVE     '0'
*IN99
C                   EVAL     SF2RRN       =
O@SFILE
C                   END
C                   END
\*  ------------------------------------------------------  End  ----
```

%FOUND(filename)

Original code

```
C                   IF          %FOUND(MDDCFG)
C ...
C ...
C                   endif
```

Changed code

```
C                   IF          (%FOUND(MDDCFG)
//*INFMBRK
C                               )
//*INFMBRK
C ...
C ...
C                   endif
```

WebTop code

```
C                   IF          (%FOUND(MDDCFG)
//*INFMBRK
C                               )
//*INFMBRK
C                               and jobtype <> 'W'
C                               OR (jobtype = 'W'
and O@SFILE<>*hival)
C ...
C ...
C                   endif
```

%EOF(filename)

Original code:

```
C                   IF          %EOF(MDDCFG)
C ...
C ...
C                   endif
```

Changed code:

```
C                   IF          (%EOF(MDDCFG)
//*INFMBRK
C                               )
//*INFMBRK
C ...
C ...
C                   endif
```

WebTop code

```
C                   IF          (%EOF(MDDCFG)
//*INFMBRK
C                               )
//*INFMBRK
C                               and jobtype <> 'W'
C                               OR (jobtype = 'W'
and O@SFILE<>*hival)
C ...
C ...
C                   endif
```

NOT %FOUND(filename)

Original code

```
C                   IF          not %found(MDDCFG)
C ...
C ...
C                   endif
```

Changed code

```
C                   IF          (NOT
%FOUND(MDDCFG)                          //*INFMBRK
C                               )
//*INFMBRK
C ...
C ...
C                   endif
```

WebTop code

```
C                   IF          (NOT
%FOUND(MDDCFG)                          //*INFMBRK
C                               )
//*INFMBRK
C                               and jobtype <> 'W'
C                               OR (jobtype = 'W'
and O@SFILE=*hival)
C ...
C ...
C                   endif
```

NOT %EOF(filename)

Original code

```
C                   IF not          %EOF(MDDCFG)
C ...
C ...
C                   endif
```

Changed code

```
C                   IF              (NOT %EOF(MDDCFG)
//*INFMBRK
C                                   )
//*INFMBRK
C ...
C ...
C                   endif
```

WebTop code

```
C                   IF              (NOT %EOF(MDDCFG)
//*INFMBRK
C                                   )
//*INFMBRK
C                                   and jobtype <> 'W'
C                                   OR (jobtype = 'W'
and O@SFILE=*hival)
C ...
C ...
C                   endif
```

%FOUND and %FOUND()

Note: When the filename is not given, the program uses the last file I/O to determine if this %found is linked to a display file, and if so, inserts the Web code.

Original code

```
C                   IF            %FOUND
C ...
C ...
C                   endif
```

Changed code

```
C                   IF            (%FOUND
//*INFMBRK
C                                  )
//*INFMBRK
C ...
C ...
C                   endif
```

WebTop code

```
C                   IF            (%FOUND
//*INFMBRK
C                                  )
//*INFMBRK
C                                  and jobtype <> 'W'
C                                  OR (jobtype = 'W'
and O@SFILE<>*hival)
C ...
C ...
C                   endif
```

%EOF and %EOF()

Note: When the filename is not given, the program uses the last file I/O to determine if this %found is linked to a display file, and if so, inserts the Web code.

Original code:

```
C                   IF        %EOF (MDDCFG)
C ...
C ...
C                   endif
```

Changed code:

```
C                   IF        (%EOF
//*INFMBRK
C                             )
//*INFMBRK
C ...
C ...
C                   endif
```

WebTop code

```
C                   IF        (%EOF
//*INFMBRK
C                             )
//*INFMBRK
C                             and jobtype <> 'W'
C                             OR (jobtype = 'W'
and O@SFILE<>*hival)
C ...
C ...
C                   endif
```

NOT %FOUND and NOT %FOUND()

Note: When the filename is not given, the program uses the last file I/O to determine if this %found is linked to a display file, and if so, inserts the Web code.

Original code

```
C                   IF          not %found(MDDCFG)
C ...
C ...
C                   endif
```

Changed code

```
C                   IF          (NOT
%FOUND(MDDCFG) //*INFMBRK
C                               )
//*INFMBRK
C ...
C ...
C                   endif
```

WebTop code

```
C                   IF          (NOT
%FOUND(MDDCFG) //*INFMBRK
C                               )
//*INFMBRK
C                               and jobtype <> 'W'
C                               OR (jobtype = 'W'
and O@SFILE=*hival)
C ...
C ...
C                   endif
```

NOT %EOF

Note: When the filename is not given, the program uses the last file I/O to determine if this %found is linked to a display file, and if so, inserts the Web code.

Original code:

```
C                   IF          not %EOF(MDDCFG)
C ...
C ...
C                   endif
```

Changed code:

```
C                   IF          (NOT %EOF(MDDCFG)
//*INFMBRK
C )
//*INFMBRK
C ...
C ...
C                   endif
```

WebTop code

```
C                   IF          (NOT %EOF(MDDCFG)
//*INFMBRK
C                               )
//*INFMBRK
C                               and jobtype <> 'W'
C                               OR (jobtype = 'W'
and O@SFILE=*hival)
C ...
C ...
C                   endif
```

OPEN

Original code

```
C                   open       mddcfg
```

WebTop code

```
C     JOBTYPE       IFNE       'W'
C                   open       mddcfg
C                   END
```

CLOSE

Original code

```
C                   close     mddcfg
```

WebTop code

```
C     JOBTYPE       IFNE      'W'
C                   close     mddcfg
C                   END
```

Notes