

Natural Engineer

Version 4.4.2

WebStar

## **Manual Order Number: NWS442-020ALL**

Copyright © August 2003, Generation Systems Ltd., East Grinstead, UK.

This document applies to Natural Engineer version 4.4.2 and to all subsequent releases.

Specifications contained herein are subject to change, and these changes will be reported in subsequent revisions or editions.

Readers' comments are welcomed. Comments may be addressed to the Documentation Department at the address on the back cover. Internet users may send comments to the following e-mail address:

[document@gensystems.com](mailto:document@gensystems.com)

## **Acknowledgements**

Microsoft, Windows, Windows NT, Visio and Visual Basic are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.

IBM, CICS, OS/390, VSE/ESA and WebSphere are either trademarks or registered trademarks of the IBM Corporation in the United States and/or other countries.

Sun, iPlanet and Solaris are either trademarks or registered trademarks of Sun Microsystems Corporation in the United States and/or other countries.

Netscape is a trademark or registered trademark of the Netscape Communications Corporation in the United States and/or other countries.

All other trademarks and registered trademarks are the sole property of their respective owners.

# TABLE OF CONTENTS

<b>ABOUT THIS MANUAL</b> .....	<b>1</b>
Purpose of this manual .....	1
Target Audience .....	1
Typographical Conventions used in this manual .....	2
How this manual is organized .....	3
Terminology .....	4
Related Literature .....	7
<b>1. CONCEPTS AND FACILITIES</b> .....	<b>9</b>
Chapter Overview.....	9
What is WebStar? .....	10
Operating Environment .....	11
Modifying Application Objects .....	19
Web Execution Process .....	31
<b>2. INSTALLATION</b> .....	<b>37</b>
Chapter Overview.....	37
Installing WebStar under Windows.....	38
Installing WebStar under OS/390 and VSE/ESA .....	40
Installing WebStar under Solaris 8 .....	47
Installing WebStar for SMARTS under OS/390 .....	50
<b>3. CONFIGURATION</b> .....	<b>55</b>
Chapter Overview.....	55
Configuration Options .....	56
Web Server .....	58
EntireX Communicator Components .....	80
Natural.....	85

**Natural Engineer WebStar**

<b>4. WEB-ENABLING A NATURAL APPLICATION .....</b>	<b>97</b>
Chapter Overview .....	97
Applying the WebStar Development Function under Windows .....	98
Applying the WebStar Development Function under OS/390 .....	104
Preparing the Web Enabled Application for Execution .....	109
Executing the Web Enabled Natural Application.....	113
Reporting Mode Considerations.....	120
Example using Sample Application: NEEEXWB.....	121
<b>5. SHORT TRANSACTION PROCESSING.....</b>	<b>143</b>
Chapter Overview .....	143
Introduction .....	144
Creating Short Transactions .....	145
Executing Short Transactions.....	175
Example Short Transaction .....	179
<b>INDEX .....</b>	<b>191</b>

# ABOUT THIS MANUAL

## Purpose of this manual

---

This manual contains the full documentation for Natural Engineer add-on component WebStar.

It describes the following topics:

- The concepts and facilities behind WebStar.
- The installation of WebStar development and run time functions.
- The configuration of WebStar run time functions.
- How to modify a Natural application with the WebStar development function, to enable the application to be executed via the Internet.
- How to create and execute Natural Short Transactions using the WebStar run time function, to enable Natural requests to be processed via the Internet.

## Target Audience

---

The target audience for this manual is intended to be any User of Natural Engineer at any level of experience.

## Typographical Conventions used in this manual

---

The following conventions are used throughout this manual:

<b>UPPERCASE TIMES</b>	Commands, statements, names of programs and utilities referred to in text paragraphs appear in normal (Times) uppercase.
<b>UPPERCASE BOLD COURIER</b>	In illustrations or examples of commands, items in uppercase bold courier must be typed in as they appear.
< >	Items in angled brackets are placeholders for user-supplied information. For example, if asked to enter <file number>, you must type the number of the required file.
<u>Underlined</u>	Underlined parts of text are hyperlinks to other parts within the online source manual. This manual was written in MS-Word 97 using the "hyperlink" feature.

The following symbols are used for instructions:

⇒	Marks the beginning of an instruction set.
□	Indicates that the instruction set consists of a single step.
1.	Indicates the first of a number of steps.

## How this manual is organized

---

This manual is organized to reflect the Concepts and Facilities, Installation/Administration and User Guide for the Natural Engineer add-on component WebStar in the following chapters:

<b>Chapter</b>	<b>Contents</b>
1	Describes the Concepts and Facilities provided by the Natural Engineer add-on component WebStar.
2	Describes the installation requirements for the Natural Engineer add-on component WebStar.
3	Describes the configuration requirements for the Natural Engineer add-on component WebStar.
4	Describes how to apply the Natural Engineer add-on component WebStar to a Natural application.
5	Describes the processes required to create and execute Natural Short Transactions using the Natural Engineer add-on component WebStar.

## Terminology

---

It is assumed that you are familiar with general Natural and mainframe terminology, as well as the terms and concepts relating to Microsoft Windows operating systems. This section explains some terms that are specific to the Natural Engineer product.

### **Analysis**

The Analysis process of Natural Engineer searches application data within the Natural Engineer Repository, according to specified Search Criteria and generates reports on the search results.

### **Application**

An Application is a library or group of related libraries, which define a complete Application. In Natural Engineer, the Application can have a one-to-one relationship with a single library of the same name, or a library of a different name, as well as related steplibs. The Application refers to all the source code from these libraries, which Natural Engineer loads into the Repository.

### **Browser**

An Internet Browser such as Microsoft Internet Explorer or Netscape.

### **Category**

Categories in Natural Engineer specify whether and how a Modification is applied to the Natural code. Valid categories are: Automatic change, Manual change, Reject the default Modification, No change to the data item, and the data item is in Generated Code.

A category is further broken down according to type of change (for example: Keyword, Literal, Data Item, Database Access, Definition).

### **Consistency**

An option in the Analysis process that causes Natural Engineer to trace an Impact through the code, using left and right argument resolution to identify further code impacted by the code found.

## **Environment**

The Environment process is the means by which Natural Engineer generates a structured view of the application code in the Natural Engineer Repository. This provides application analysis reports and inventory information on the application and is used as the basis for Impact Analysis.

## **Exception**

An Exception is an Item identified as impacted that does not require a Modification. Where there are a few similar Exception Items, they can be treated as Exceptions, and rejected in the Modification review process. Where there are many similar (therefore not Exceptions), consideration should be given to changing the Search Criteria so they are not identified as impacted in the first place.

## **Generated Code**

This is code which has been generated by a Natural code generator, such as Construct, and which is not normally modified directly in the Natural editor.

## **Impact**

An Impact is an instance of a Natural code Item; e.g., data item or statement (a “hit” scored by the Analysis process) that matches the defined Search Criteria used in the Analysis process.

## **Iteration**

An Iteration is one examination cycle of a field identified according to the specified Search Criteria. For example, one Iteration is reading the field right to left. Multiple Iterations are performed when the option of ‘Consistency’ or Multi Search is requested for Analysis, and Natural Engineer performs as many Iterations as necessary to exhaust all possibilities of expressing and tracing the field, and can be limited by a setting in the NATENG.INI file.

## **Library**

A single library of source code, which exists in the Natural system file.

## **Natural Engineer WebStar**

### **Modification**

A Modification is a change suggested or made to an object or data item resulting in the required compliance of that object or data item. Modifications in Natural Engineer are classified according to Category and Type.

### **Presentation Split Process**

The Presentation Split Process is a sub-function of the Object Builder function that removes screen I/O statements from current application objects and places them in generated subprograms.

### **Soft Link**

A Soft Link is where a link between two objects has been defined using an alphanumeric variable rather than a literal constant.

### **Technical Split Process**

The Technical Split Process is a sub-function of the Object Builder function that results in the encapsulation of each database access within the application, into a sub-program so that the application is separated into 'presentation and logic' and 'database access'.

### **Type**

The Type of Modification available, for example: Data Item, Keyword and Literal.

### **TLM**

Text Logic Members are used to contain the code required to support inclusion of common code into the application. An example of this is the code to include into an application before updating a database.

## Related Literature

---

The complete set of Natural Engineer manuals consists of:

**1 Natural Engineer Concepts and Facilities (NEE442-006ALL)**

The Concepts and Facilities manual describes the many application systems problems and solutions offered by Natural Engineer, providing some guidelines and usage that can be applied to Natural applications.

**2 Natural Engineer Release Notes (NEE442-008ALL)**

The Release Notes describe all the information relating to the new features, upgrades to existing functions and documentation updates that have been applied to Natural Engineer.

**3 Natural Engineer Installation Guide (NEE442-010ALL)**

The Installation Guide provides information on how to install Natural Engineer on both PC and mainframe platforms.

**4 Natural Engineer Administration Guide (NEE442-040WIN)  
Natural Engineer Administration Guide (NEE442-040MFR)**

The Administration Guide provides information on all the various control settings available to control the usage of the different functions within Natural Engineer.

**5 Natural Engineer Application Management (NEE442-020WIN)  
Natural Engineer Application Management (NEE442-020MFR)**

The Application Management manual describes all the functions required to add Natural applications into the Repository.

**6 Natural Engineer Application Documentation (NEE442-022WIN)  
Natural Engineer Application Documentation (NEE442-022MFR)**

The Application Documentation manual describes all the available functions to document a Natural application within the Repository. These functions will help enhance / supplement any existing systems documentation such as BSD / CSD / Specifications etc.

**7 Natural Engineer Application Analysis and Modification (NEE442-023WIN)  
Natural Engineer Application Analysis and Modification (NEE442-023MFR)**

The Application Analysis and Modification manual describes all the available functions to carry out analysis of Natural applications; including basic keyword searches. The modification process is described and detailed to show how it can be applied to modify single selected objects within a Natural application, or the entire Natural application in one single execution.

## **Natural Engineer WebStar**

**8 Natural Engineer Application Restructuring (NEE442-024WIN)  
Natural Engineer Application Restructuring (NEE442-024MFR)**

The Application Restructuring manual describes the analysis and modification functionality required to carryout some of the more sophisticated functions such as Object Builder.

**9 Natural Engineer Utilities (NEE442-080WIN)  
Natural Engineer Utilities (NEE442-080MFR)**

The Utilities manual describes all the available utilities found within Natural Engineer and, when and how they should be used.

**10 Natural Engineer Reporting (NEE442-025ALL)**

The Reporting manual describes each of the reports available in detail, providing report layouts, how to trigger the report and when the report data becomes available. The various report-producing mediums within Natural Engineer are also described.

**11 Natural Engineer Batch Processing [Mainframes] (NEE442-026MFR)**

The Batch Processing manual describes the various batch jobs (JCL) and their functionality.

**12 Natural Engineer WebStar (NWS442-020ALL)**

The WebStar manual describes the concepts and facilities, installation and configuration options, how to web enable a Natural application and how to create and execute Natural Short Transactions using the Natural Engineer add-on component WebStar.

**13 Natural Engineer WebStar Release Notes (NWS442-008ALL)**

The Release Notes describe all the information relating to the new features, upgrades to existing functions and documentation updates that have been applied to the Natural Engineer add-on component WebStar.

**14 Natural Engineer Messages and Codes (NEE442-060ALL)**

The Messages and Codes manual describes the various messages and codes produced by Natural Engineer.

# CONCEPTS AND FACILITIES

## Chapter Overview

---

This chapter provides an overview of the concepts and facilities provided by the Natural Engineer add-on component WebStar.

The topics covered are:

- [What is WebStar?](#)
- [Operating Environment](#)
- [Modifying Application Objects](#)
- [Web Execution Process](#)

## What is WebStar?

---

WebStar is an add-on component to Natural Engineer, which provides the functionality to web-enable a Natural application. There are two levels of functionality to WebStar:

### **1. Modifying a Natural application.**

Using Natural Engineer, a Natural application is extracted and loaded into the Repository. Impact Analysis is executed using a supplied pre-defined set of Impact search criteria. After the Impact Analysis results have been checked, modification can be applied to the application.

The WebStar process identifies all presentation layer logic within an application and encapsulates it within objects that can communicate with a web browser. Any existing business and application logic is unaffected and the database accesses are unchanged.

### **2. Executing the modified Natural application using the Internet.**

Run-time support is provided for the web enabled Natural application in the form of web server components and a Natural run-time library. The modified Natural application requires to be catalogued with a steplib set to include the run-time objects (these provide the control mechanism between the Natural server session and the web server via EntireX Broker).

A stub program needs to be added to the modified application to act as the start-up object for the modified application, along with some minor administration to the server environment. The web enabled application can now be executed from a web browser.

*Note: After an application has been modified using WebStar, the application can be executed from a web browser, or, from its current Natural environment.*

## Operating Environment

---

The operating environment consists of several key individual components, which when brought together provide the necessary link between your Natural application and the Web.

The following components are now reviewed:

1. [Browser Communication](#)
2. [Web Server Communication to Natural](#)
3. [Interface Software](#)
4. [Transaction Processing](#)
5. [Web Pages](#)
6. [Overview](#)

# 1

## Natural Engineer WebStar

### Browser Communication

A browser requires a web server to communicate to the backend application or static files. The web server can exist on a PC, a Unix machine or a mainframe. The web server provides the ability to execute objects automatically and to store files for display in a browser.

The following Figure 1-1 illustrates the relationship between a web server and a user.

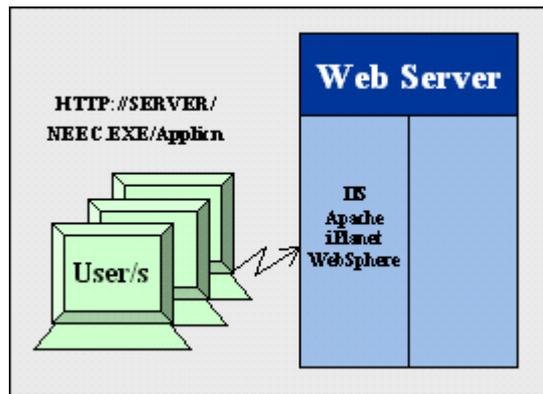


Figure 1-1 Relationship between a web server and a user

## Web Server Communication to Natural

To access the backend application, in this case Natural objects, there needs to be a communication method in place. WebStar uses EntireX Broker (a high performance message server) to provide the communication mechanism between the web server and Natural. The standard communication will involve the passing of data and navigational controls (such as PF Keys) between the web pages and the Natural application.

The following Figure 1-2 illustrates the use of EntireX Broker in the WebStar solution.

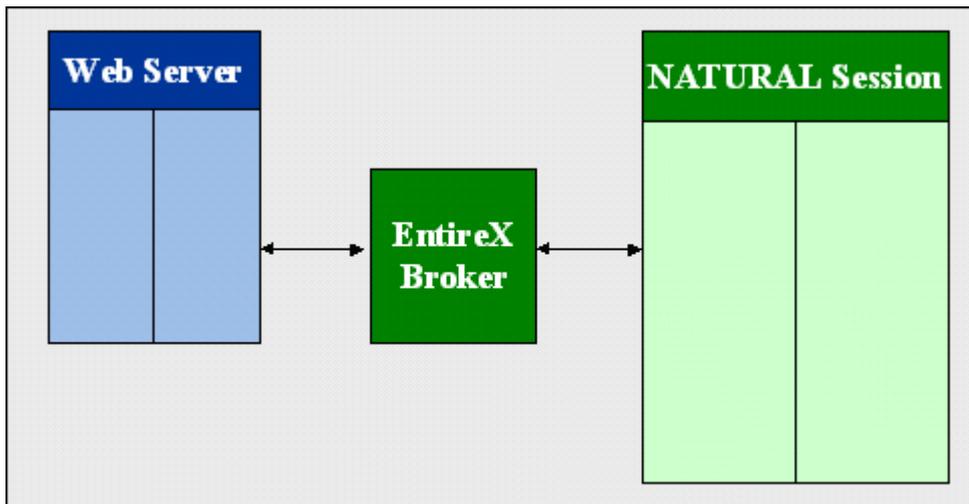


Figure 1-2 Use of EntireX Broker in the WebStar solution

## Interface Software

In order to make use of EntireX Broker to communicate between the web server and the Natural application, WebStar utilizes two types of controlling objects:

- (1) Interface program for the web server (NEEC)
- (2) Natural run time objects for the Natural application to use.

Both sets of controlling objects provide the data and/or navigational control information from the web server to the Natural application. And then, any resultant actions are relayed back from the Natural application to the web server.

The following Figure 1-3 illustrates the location of the interface software.

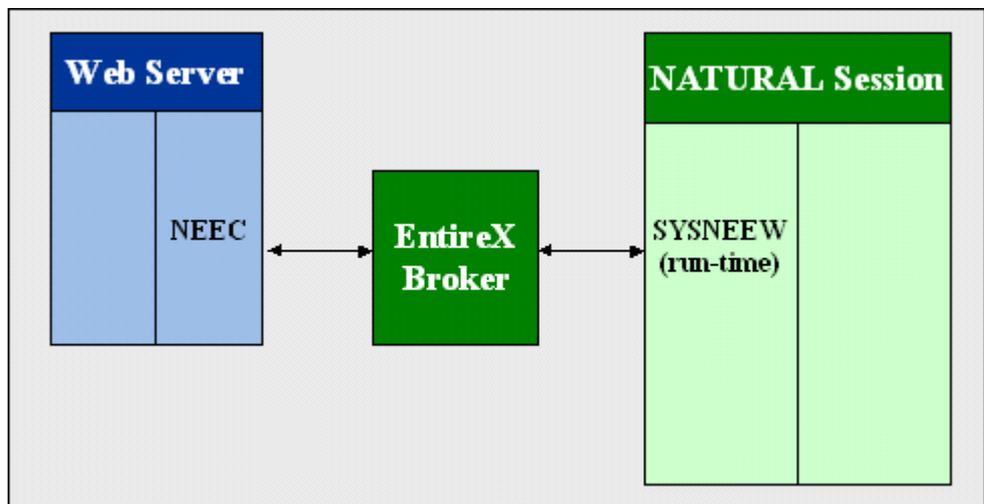


Figure 1-3 Location of interface software

## Transaction Processing

Natural applications executed using WebStar via a web browser will initiate a Natural session for each web user. This provides the flexibility and security to handle 'long transactions' (i.e., database accesses that exist across screens).

To achieve this, WebStar makes use of various Attach Managers (ATM) to both initiate and maintain individual Natural sessions. The ATM's available are:

<b>Attach Manager Version</b>	<b>Applicable Platform</b>
EntireX Attach Service	CICS OS/390
Windows Attach Manager (NEEATM)	Windows and UNIX
Mainframe Attach Manager (NEEATM)	Mainframe sessions running under Com-Plete OS/390

The following Figure 1-4 illustrates the relationship between EntireX Broker and the Attach Manager (NEEATM).

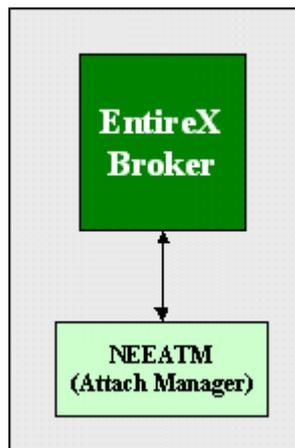


Figure 1-4 Attach Manager (NEEATM)

## Transaction Types

The WebStar solution supports two types of transaction:

### 1. Short Transactions

Short transactions are when a request is sent to the Natural Server, the Natural Server services the transaction and returns the results back to the web browser. The short transaction is now over. For example:

Request a list of application objects. The list objects request is sent to the Natural Server, the Natural Server services the transaction and returns the results back to the web browser. The transaction is now complete.

*Note: The short transaction data can be returned in HTML/XML, Microsoft Excel, Microsoft Word or Adobe PDF formats.*

### 2. Long Transactions

Long transactions are issued when an application is running, where a request is sent to the Natural session, the Natural session services the transaction and returns the results back to the web browser, and waits for the next request. The transaction process continues until the application terminates. For example:

An application is started and a menu option has been selected. The menu option request is sent to the Natural session, the menu option is processed and the Natural session returns the result - the screen for menu option requested.

## Web Pages

The WebStar modification process will generate the web pages for the presentation layers within an application in XSL and HTML formats. These web pages will be stored on the web server being used to execute the application via the Internet. They can also be placed at other web locations, for example, an XML database e.g., Tamino.

The run time data will be passed within the application using XML documents controlled by the NEEC program and EntireX Broker.

The following Figure 1-5 illustrates the relationship between the NEEC web server interface program and the XSL/HTML web pages.

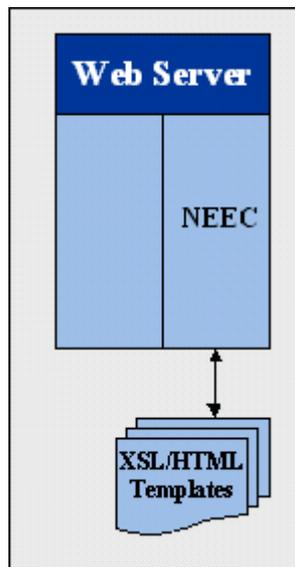


Figure 1-5 XSL/HTML web pages

# 1

## Natural Engineer WebStar

### Overview

The operating environment consists of several individual components, which when combined together enable a Natural application to be executed across the Internet.

Use of standard communication protocol means that there is no requirement for all the components to reside on one single machine, resulting in minimal effort to integrate the WebStar solution.

The following Figure 1-6 illustrates the complete WebStar operating environment.

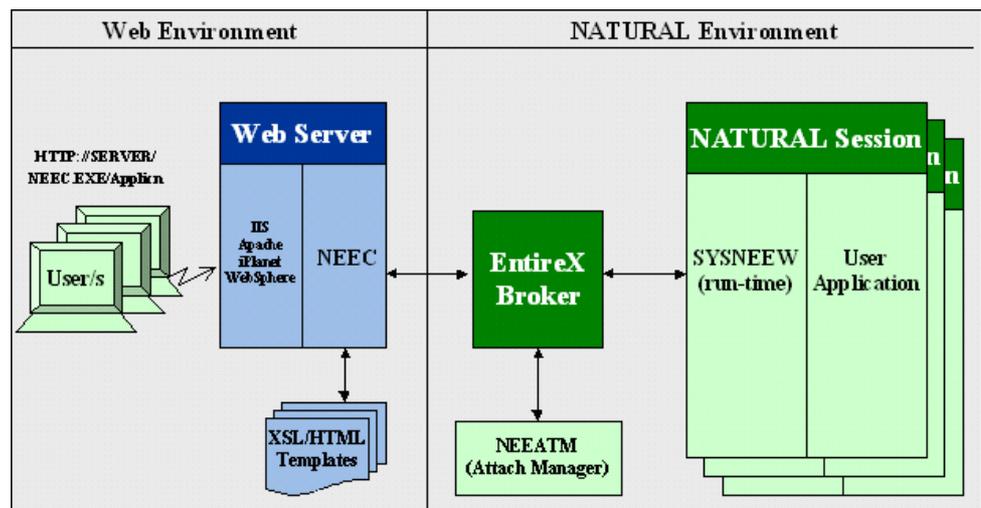


Figure 1-6 WebStar operating environment

## Modifying Application Objects

---

This section describes the process used by the WebStar add-on component to modify a Natural application using Natural Engineer and the various objects created.

### Natural Engineer Process

To web-enable a Natural application, the following steps are required using Natural Engineer with the WebStar add-on component:

1. Extract the Natural application creating a load file containing a neutral view of the application source code.
2. Review the Extract process by checking the Extract error log file for any extract errors and producing the Missing Objects report.
3. Load the neutral view of the application source code into the Natural Engineer Repository.
4. Review the Load process by checking the Load error log file and producing any of the Application Reports, for example the Source Code Summary report.
5. Execute Impact Analysis using the supplied pre-defined Impact search criteria. This will identify the presentation layer logic within the application objects.
6. Review the Impact Analysis results.
7. Apply Modification to the application objects. This will modify the objects containing the presentation layer logic and generate new sub-programs to control the presentation layer. Also generated are the XSL and HTML web pages, which are used when, executing the modified application over the Internet.

### Modified Natural Objects

The Natural application is modified so that it can be executed over the Internet, or run as normal using a Natural session, i.e., after modification the system can still be executed under Natural the same as before the modification.

The modification process will:

# 1

## Natural Engineer WebStar

- Generate a subprogram for each presentation statement in the application source code. Presentation statements include any I/O type statements, such as: WRITE, DISPLAY, INPUT USING MAP.
- Existing presentation statements are retained and processing statements to call the appropriate subprogram for the web page display are added. These are controlled by a conditional block checking the system variable \*CONVID, i.e., they are only executed if the value of \*CONVID is greater than 0 (non-Natural session).
- Any 'SET KEY' statements are recognized and carried forward to the appropriate sub-program.

The following Figure 1-7 illustrates sample source code for program XX001P01 before modification.

```
0010 DEFINE DATA GLOBAL USING XX000G00
0020 LOCAL USING XX001L01
0030 *
0040 END-DEFINE
0050 *
0060 SET KEY ALL
0070 *
0080 REPEAT
0090 *
0100 INPUT USING MAP "XX001M01"
0110 RESET #L-MESSAGE
0120 *
0130 DECIDE ON FIRST VALUE OF *PF-KEY
0140 *
0150 VALUE "PF12", "PF24"
0160     PERFORM XXEXIT
0170 VALUE 'ENTR'
0180     IF #M-OPTION = "P"
0190         FETCH "XX002P01"
0200     END-IF
0210     IF #M-OPTION = "S"
0220         FETCH "XX003P01"
0230     END-IF
0240     MOVE "INVALID OPTION SELECTED" TO #L-MESSAGE
0250 NONE VALUE
0260     MOVE "INVALID PF KEY PRESSED" TO #L-MESSAGE
0270 END-DECIDE
0280 *
0290 END-REPEAT
0300 END
```

Figure 1-7 Sample source code for program XX001P01 before modification

The following Figure 1-8 illustrates the sample source code for program XX001P01 after modification (modified code shown in bold).

```

0010 DEFINE DATA GLOBAL USING XX000G00
0020 LOCAL USING XX001L01
0030 *
0040 LOCAL USING NEESTD-A /* NEE MODIFIED
0050 LOCAL USING NEECON-L /* NEE MODIFIED
0060 LOCAL USING NEEOBJ-L /* NEE MODIFIED
0070 LOCAL USING NEESYS-L /* NEE MODIFIED
0080 END-DEFINE
0090 *
0100 SET KEY ALL
0110 *
0120 MOVE #@NEE-PFKEY-ACTIVE TO #@NEE-KEY-PGMSSENSE-STATUS(*) /* NEE MODIFIED
0130 RESET #@NEE-KEY-CMD-STATUS(*) #@NEE-KEY-COMMAND(*)/* NEE MODIFIED
0140 REPEAT
0150 *
0160 INCLUDE NEEQ /* NEE MODIFIED
0170 INPUT USING MAP "XX001M01"
0180 IF *CONVID GT 0 /* NEE MODIFIED
0190 CALLNAT 'XX001PN1' #@NEE-STD-PARMS /* NEE MODIFIED
0200 #L-MESSAGE #M-OPTION #@NEE-SYS-PROGRAM #@NEE-SYS-DATE #@NEE-SYS-USER /*
0210 END-IF /* NEE MODIFIED
0220 RESET #L-MESSAGE
0230 *
0240 DECIDE ON FIRST VALUE OF *PF-KEY
0250 *
0260 VALUE "PF12", "PF24"
0270     PERFORM XXEXIT
0280 #@NEE-STD-PARMS /* NEE MODIFIED
0290 VALUE 'ENTR'
0300     IF #M-OPTION = "P"
0310         FETCH "XX002P01"
0320     END-IF
0330     IF #M-OPTION = "S"
0340         FETCH "XX003P01"
0350     END-IF
0360     MOVE "INVALID OPTION SELECTED" TO #L-MESSAGE
0370 NONE VALUE
0380     MOVE "INVALID PF KEY PRESSED" TO #L-MESSAGE
0390 END-DECIDE
0400 *
0410 END-REPEAT
0420 END
0430

```

Figure 1-8 Sample source code for program XX001P01 after modification

The following Figure 1-9 illustrates sample source code for the generated subprogram XX001PN1 used to control the presentation layer when executed over the Internet.

```

0010 * Subprogram: XX001PN1
0020 *****
0030 * Created by NEE on 2002-03-01 at 13:54:00.0
0040 * Created from XX001P01 from line range 0100-0109
0050 *****
0060 DEFINE DATA
0070 PARAMETER USING NEESTD-A
0080 PARAMETER
0090 01 #XX001PN1-PDA
0100 02 #L-MESSAGE (A70)
0110 02 #M-OPTION (A1)
0120 02 #@NEE-SYS-PROGRAM (A8)
0130 02 #@NEE-SYS-DATE (A8)
0140 02 #@NEE-SYS-USER (A8)
0150 *
0160 LOCAL USING NEESTD-L
0170 LOCAL
0180 01 #ETB
0190 02 #ETB-IO (A250/1:60)
0200 02 REDEFINE #ETB-IO
0210 03 #ETB-SB (A250/1:60)
0220 02 REDEFINE #ETB-IO
0230 03 #ETB-RB (A250/1:60)
0240 02 REDEFINE #ETB-IO
0250 03 #ETB-DATA-BLOCK
0260 04 #L-MESSAGE (A70)
0270 04 #M-OPTION (A1)
0280 04 #@NEE-SYS-PROGRAM (A8)
0290 04 #@NEE-SYS-DATE (A8)
0300 04 #@NEE-SYS-USER (A8)
0310 02 #ETB-EB (A40)
0320 02 #ETB-EC (A08)
0330 *
0340 01 #HTML-FB (A86/1:165) INIT
0350 (1) < '#L-MESSAGE A070
0360 (2) < '#M-OPTION A001
0370 (3) < '#@NEE-SYS-PROGRAM A008
0380 (4) < '#@NEE-SYS-DATE A008
0390 (5) < '#@NEE-SYS-USER A008
0400 *
0410 END-DEFINE
0420 *
0430 MOVE BY NAME #XX001PN1-PDA TO #ETB-DATA-BLOCK
0440 MOVE 'XX001M01' TO #@NEE-MAP-NAME
0450 MOVE TRUE TO #@NEE-OUTPUT-PFKEYS
0460 *

```

```
0470 CALLNAT 'NEEDAT-N' #@NEE-STD-PARMS #ETB #HTML-FB(*)
0480 *
0490 CALLNAT 'NEEETB-N' 'CONVERSE' #ETB
0500 *
0510 CALLNAT 'NEEDCP-N' #@NEE-STD-PARMS #ETB-RB(*) #HTML-FB(*)
0520 *
0530 CALLNAT 'NEECMD-N' #@NEE-STD-PARMS
0540 *
0550 RESET #@NEE-FOCUS-FIELD #@NEE-REINPUT-LINE
0560 IF #@NEE-PF-KEY EQ 'PA1' OR EQ 'PA2' OR EQ 'CLR'
0570     ESCAPE ROUTINE
0580 END-IF
0590 *
0600 MOVE #ETB.#M-OPTION TO #XX001PN1-PDA.#M-OPTION
0610 *
0620 END
0630
```

Figure 1-9 Generated subprogram XX001PN1

The generated subprogram controls all the information required to pass data between the Natural session and the web browser.

Overview of subprogram functions:

1. Receives user data for the browser.
2. Converts the user data to XML format.
3. Calls the EntireX Broker with the data buffer and awaits response.
4. After receiving response, converts the received user data in XML format back to Natural format.
5. Sets the Natural PF Key to the browser command received.
6. Continues in Natural session until the next presentation statement encountered.

## Generated Web Pages

For each presentation statement within an application, an XSL file is created containing the presentation layout displayed in the web browser. The XSL file complements the data handling and processing control handled by the generated subprogram.

The XSL files contain the static display information and also the data items to be used at run time. The data that populates the data items in the XSL file is passed from the Natural session after being converted to XML format. At run time the XML and the XSL data is merged and presented as an HTML page in the web browser.

*Note: The XML and XSL merge can occur, either at the server side or client side depending on the environment being used.*

The XSL files can be one of two types:

1. [Single View Data](#)
2. [Multiple View Data](#)

## Single View Data

This is used for single occurrence I/O data, such as the input of parameters into a called program, or when a map is being displayed.

Example Natural statements:

```
INPUT USING MAP 'MAP1'
```

At run time the XML data is created by the generated subprogram for each presentation statement. The XML data contains a reference to the XSL file with which the data is to be merged. The web browser (client-side) or web server (server-side) will locate the appropriate XSL file, replacing the data items with the data supplied by the XML file. Once the data has been merged, the final HTML page is displayed in the web browser.

*Note: All special characters used for data item names within Natural, for example '#' or '-' (hyphen), are removed as these characters are not valid within XSL.*

*If after the removal of special characters, the data item has a leading numeric character, then the data item will be prefixed with 'NEE'. For example, data item #001-OPTION will become NEE001OPTION.*

*Otherwise, the data item names are as found in the Natural objects.*

The following Figure 1-10 illustrates some sample blocks from the XSL file for single view data.

```

<td colspan="30" >
Welcome to the Hospital System
</td>
<td colspan="16">&nbsp;</td>
<td colspan="8">
<xsl:value-of select="NEESYSDATE" />
</td>
</tr>
<tr>
<td colspan="8">
<xsl:value-of select="NEESYSUSER" />
</td>
<td colspan="16">&nbsp;</td>
<td colspan="30" >

<tr>
<td colspan="25">&nbsp;</td>
<td colspan="1">
<xsl:variable name="MOPTION" select="MOPTION" />
<input type="TEXT" name="MOPTION" value="{ $MOPTION}" maxlength="1" size="1"
id="FA=A001 " />
</td>
<td colspan="6">&nbsp;</td>
<td colspan="28" >
Please Enter Required Option
</td>
</tr>

<tr>
<td colspan="58">&nbsp;</td>
<td colspan="15" >
PF12/24 to Exit
</td>
</tr>
<tr>
<td colspan="3">&nbsp;</td>
<td colspan="70">
<xsl:value-of select="LMESSAGE" />
</td>
</tr>
</table></center>
</div>
<hr /><div align="center"><center>
<table border="0" cellspacing="0" cellpadding="0">
<tr><td><input type="SUBMIT" name="NEEPFKEY" value="ENTR" /></td>
<td>
<input type="SUBMIT" name="NEEPFKEY" value="PF01" id="KEY=PF1" />
</td>

```

Figure 1-10 Sample blocks for the XSL file for single view data

## Concepts and Facilities

# 1

Block one contains the heading line for a map which contains two data items NEESYSDATE and NEESYSUSER which will be replaced to show the \*DATE and \*USER data from the XML file.

Block two contains an input line from the map where the input data item MOPTION will have a value of \$MOPTION passed back to Natural via the XML file.

Block three shows the message line (LMESSAGE) and the start of the PF Key's for the map.

## Multiple View Data

This is used for multiple occurrence I/O data such as that produced for reports when using WRITE/DISPLAY statements.

Example Natural statements:

```
WRITE EMPLOYEES.PERSONNEL-ID EMPLOYEES.NAME  
DISPLAY VEHICLES.MAKE VEHICLES.MODEL VEHICLES.COLOR
```

At run time the XML data is created by the generated subprogram for each presentation statement.

The XSL file will contain the data to be repeated within structured blocks referenced by start and end tags. These tags will be used by the XSL file to repeatedly merge the XML data for the block, until all the XML data for the block has been used up.

The structured blocks represent the WRITE and DISPLAY statements from the original source code. The XML data contains a reference to each structured block within the XSL file, so that the correct repeating data is merged.

The web browser (client-side) or web server (server-side) will locate the appropriate XSL file, replacing the data items with the data supplied by the XML file. Once the data has been merged, the final HTML page is displayed in the web browser.

*Note: All special characters used for data item names within Natural, for example '#' or '-' (hyphen), are removed as these characters are not valid within XSL.*

*If after the removal of special characters, the data item has a leading numeric character, then the data item will be prefixed with 'NEE'. For example, data item #001-OPTION will become NEE001OPTION.*

*Otherwise, the data item names are as found in the Natural objects.*

The following Figure 1-11 illustrates some sample blocks from the XSL file for multiple view data.

```

<xsl:for-each select="NEE_DISPLAY_001">
<tr>
<td><xsl:value-of select="OBJLONGNAME" /></td>
<td><xsl:value-of select="STTCOUNT" /></td>
<td><xsl:value-of select="STTTOTAL" /></td>
<td><xsl:value-of select="STTAVER" /></td>
<td><xsl:value-of select="STTMAX" /></td>
<td><xsl:value-of select="REPORTCOUNT" /></td>
<td><xsl:value-of select="STRUCTCOUNT" /></td>
</tr>
</xsl:for-each>

<xsl:for-each select="NEE_OBJECT-TYPE_001">
<tr>
<td><xsl:value-of select="OBJLONGNAME" /></td>
<td><xsl:value-of select="TOTALOBJECTS" /></td>
<td><xsl:value-of select="TOTALLINES" /></td>
<td><xsl:value-of select="AVERAGELINES" /></td>
<td><xsl:value-of select="LARGESTLINES" /></td>
<td><xsl:value-of select="REPORTCOUNT" /></td>
<td><xsl:value-of select="STRUCTCOUNT" /></td>
</tr>
</xsl:for-each>

```

Figure 1-11 Sample blocks for the XSL file for multiple view data

Block one shows a structured block of repeating data, which is being displayed in the main body of a report. The structured block contains the following:

- Start tag: '<xsl:for-each select="NEE\_DISPLAY\_001">'
- Set of data items that are required for each report line, i.e., OBJLONGNAME, STTCOUNT, STTTOTAL, STTAVER, STTMAX, REPORTCOUNT and STRUCTCOUNT.
- End tag: '</xsl:for-each>'

This structured block will have XML data that has a reference of 'NEE\_DISPLAY\_001' merged into it until the end of that data.

Block two shows a structured block used to show sub totals for each object type on the report. The structured block contains the following:

# 1

## Natural Engineer WebStar

- Start tag: '<xsl:for-each select="NEE\_OBJECT\_TYPE\_001">'
- Set of data items that are required for each report line, i.e., OBJLONGNAME, TOTALOBJECTS, TOTALLINES, AVERAGELINES, LARGESTLINES, REPORTCOUNT and STRUCTCOUNT.
- End tag: '</xsl:for-each>'

This structured block will have XML data that has a reference of 'NEE\_OBJECT\_TYPE\_001' merged into it until the end of that data.

## Web Execution Process

---

After the Natural application has been web enabled, the application can be executed using a Natural session (the same as the original pre-modified application), or via the Internet.

The WebStar add on component comes supplied with a set of HTML pages that provide both a security front end and processing mechanism to initiate and open the web enabled application. These will be typically located in the root folder 'NEE' on the web server.

Once the operating environment has been configured, the web enabled Natural application can be executed by using a web browser and specifying a URL.

For example: **<http://www.myweb.com/nee/webstar.html>**.

*Note: The web page reference 'webstar.html' must be included as part of the URL.*

## Web Execution Overview

The following is a high level overview of the process used to execute a web enabled Natural Application over the Internet using the WebStar solution.

1. Access to the application is made through the WEBSTAR.HTML page, which is a logon page for WebStar.
2. A user-id and password are input, and an application selected. The user-id and password are validated using the web server interface program NEEC.EXE, to communicate with EntireX Broker.
3. Once validated, this is treated as a new request and the Attach Manager is invoked to initiate a new Natural Session. The Attach Manager will reference the correct Natural Parameter module for the application to use.
4. Once the Natural session has started, the application first performs an initialization process, setting entries in the transportation process. These include Broker-id, Server name, Service name and Service.
5. At the first presentation statement within the application, the process will recognize that the application is being executed via the Internet by the setting found in \*CONVID. This will result in the generated subprogram being invoked (as opposed to the Natural map) and setting up the required screen data in XML format.
6. Once all the data has been formatted, EntireX Broker is called with the 'CONVERSE' option. This will hold the Natural session until it is released by the User, or non-activity timeouts are invoked.
7. EntireX Broker will now pass the XML data to the web server interface program NEEC.EXE. Depending on the environment being used, the web server interface program will either merge the XML data and XSL file at the web server (server-side), or modify the XML data with the correct XSL file name to be used in the merge and send it to the web browser (client-side) for the merge to take place.
8. The resulting HTML page is now displayed and processing waits for the next user action.

9. After the next user response, the web server interface program NEEC.EXE formats any XML data from the current page and passes it to EntireX Broker. EntireX Broker locates the correct Natural session and returns the XML data back to the generated subprogram. This is converted back to Natural data.
10. Control is returned to the application object and processing continues until the next presentation statement is encountered.

## Web Execution Example

The following series of illustrations show the web execution process. The web enabled application is the supplied sample application NEEEXWB.

The following Figure 1-12 illustrates the URL used to start the web execution process.

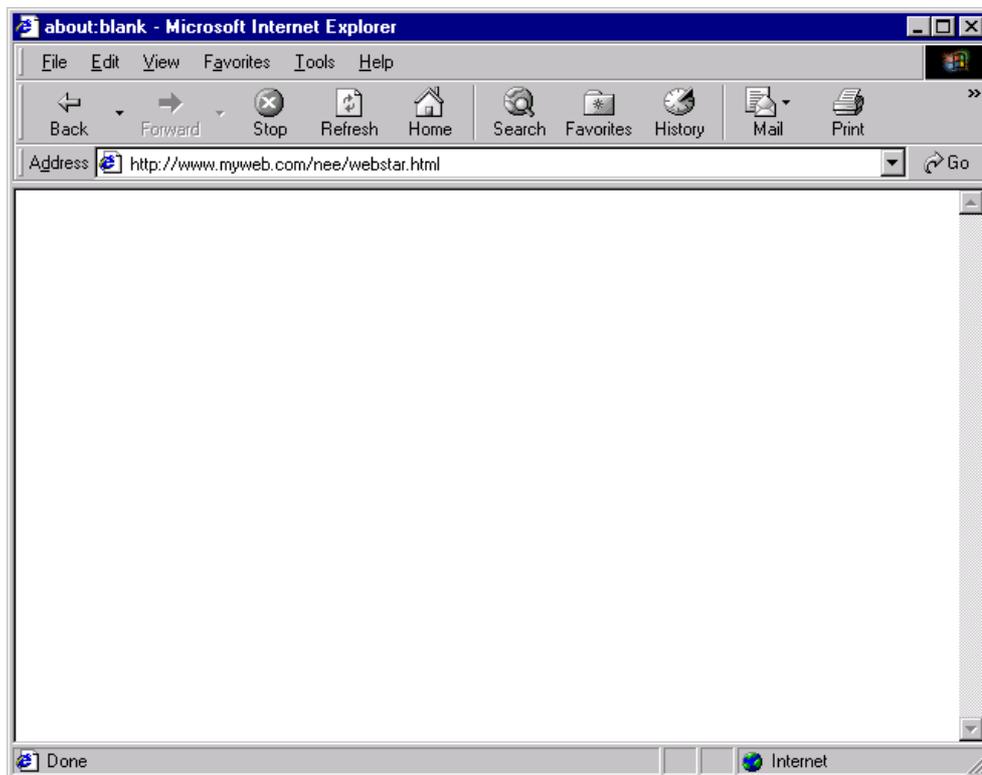


Figure 1-12 Start up URL

# 1

## Natural Engineer WebStar

The following Figure 1-13 illustrates the WebStar Logon page.

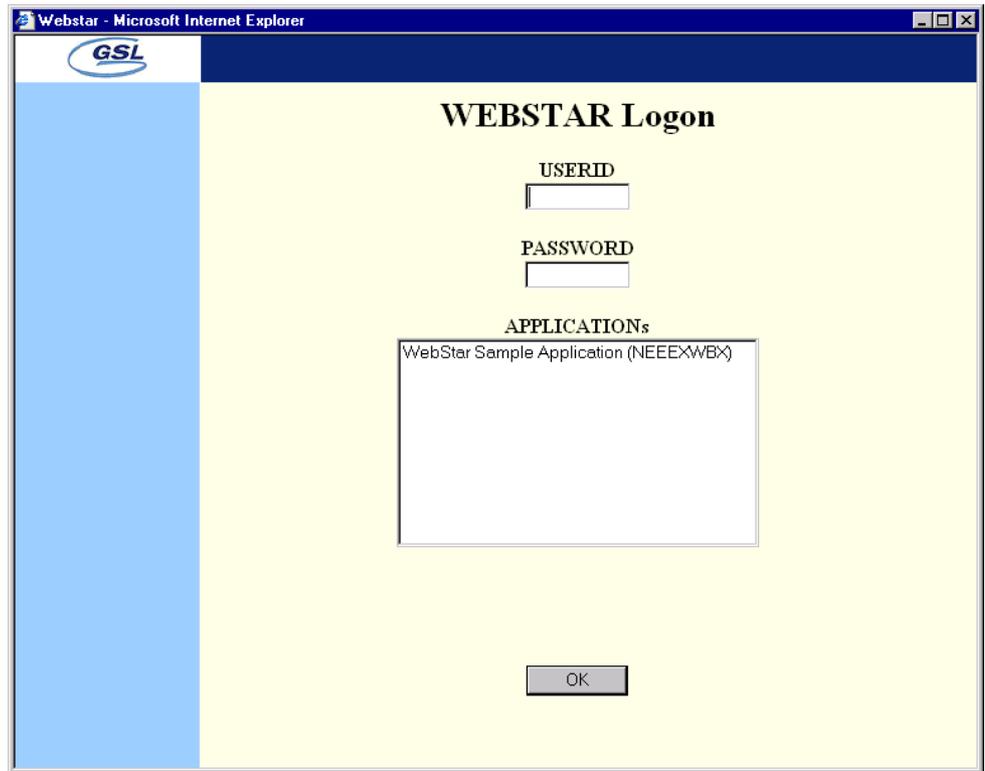


Figure 1-13 WebStar Logon page

The following Figure 1-14 illustrates the WebStar Logon page with details entered.

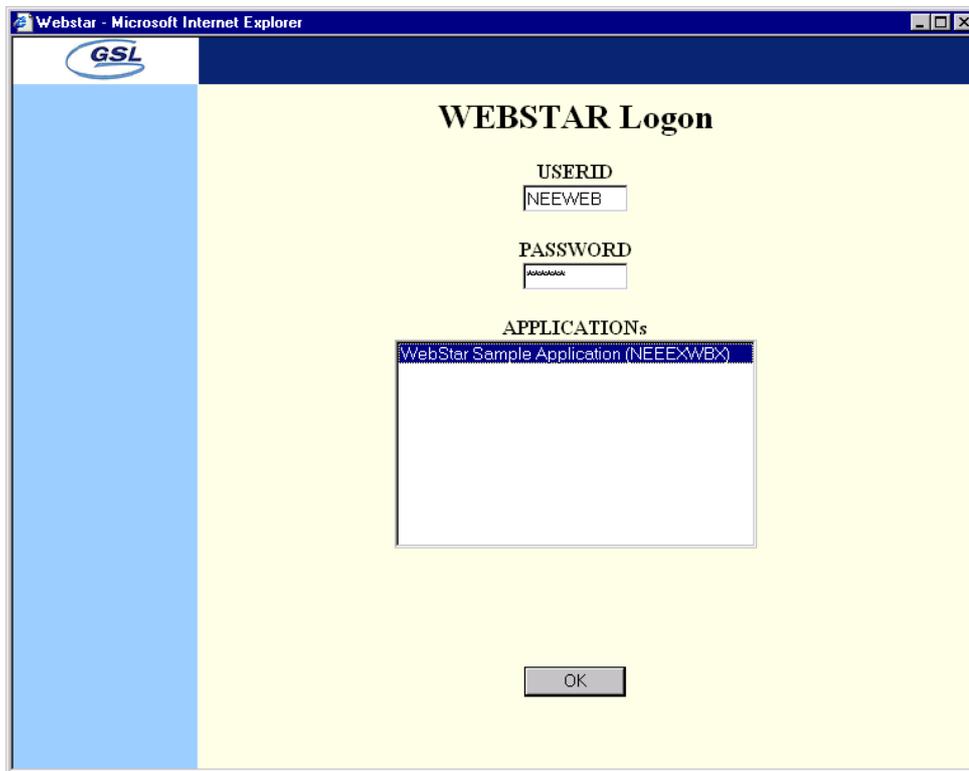


Figure 1-14 WebStar Logon page with details entered

*Note: The normal User-id and password for the application are entered as per site standards. These can be validated using the EntireX Security functionality.*

# 1

## Natural Engineer WebStar

The following Figure 1-15 illustrates the main screen for application NEEEXWB.

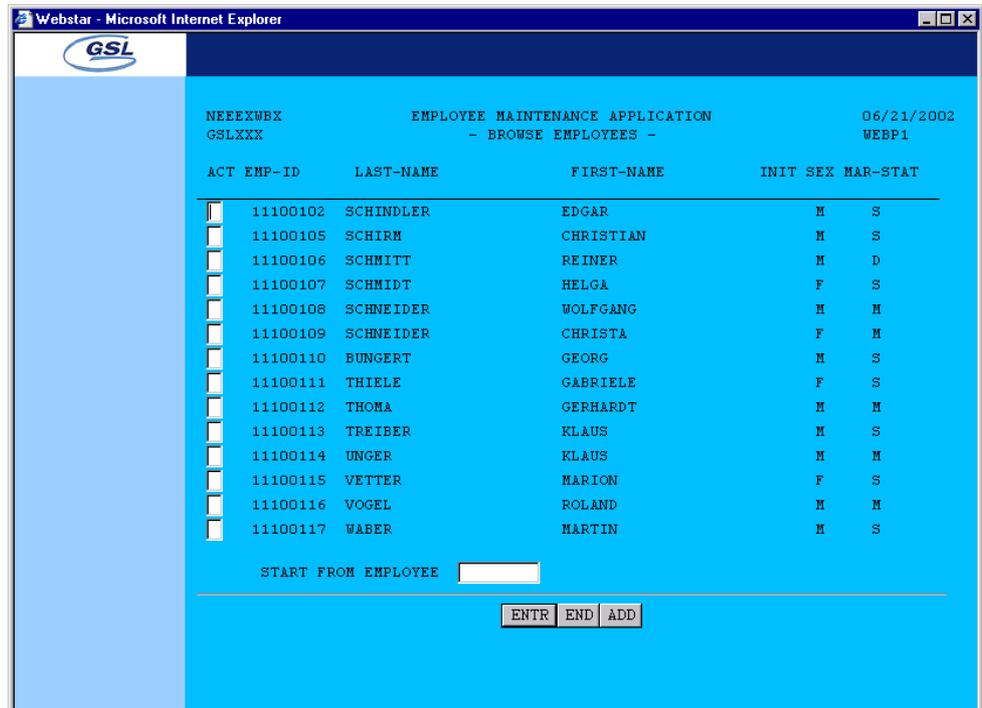


Figure 1-15 Main menu for application NEEEXWB

# INSTALLATION

## Chapter Overview

---

This chapter explains how to install the Natural Engineer add-on component WebStar.

The Natural Engineer add on component WebStar consists of two levels of functionality:

1. Modifying a Natural application using the standard Natural Engineer Extract, Load, Impact and Modification processes. This forms the WebStar development function.
2. Executing the web enabled application via the Internet. This forms the WebStar run time function.

The topics covered are:

- [Installing WebStar under Windows](#)
- [Installing WebStar under OS/390 and VSE/ESA](#)
- [Installing WebStar under Solaris 8](#)
- [Installing WebStar for SMARTS under OS/390](#)

## 2

### Natural Engineer WebStar

## Installing WebStar under Windows

---

This section describes the installation required for a single PC running under a Windows operating system.

This will install the WebStar development function and the WebStar run time components.

### Prerequisites

The following products must be installed before you install Natural Engineer WebStar:

- Windows 2000 Professional, Windows XP Professional or Windows 2000 Server.
- Natural version 5.1.1 Patch Level 5 or above.
- Adabas version 3.1.1 or above.

## Installation Process

The InstallShield process on the Natural Engineer product CD controls the installation process.

The following summarizes the new WebStar run time components added:

- Add the new predefined Impact search criteria file WEB.IRE to the DATA folder that is installed in the X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2 folder (where X: is the directory on which Natural Engineer was installed).
- Add new FNAT run time library SYSNEEW. This library is utilized when running the modified application via the Internet.
- Add new FUSER libraries:
  - NEEEXWB - contains example Natural application objects.
  - NEEEXST - contains example Natural Short Transaction objects.
- Add new WEB folder to X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2 folder (where X: is the directory on which Natural Engineer was installed). This contains:
  - NEE folder containing the base WebStar HTML files and style sheets used at run time for web enabled applications. Also contains:
    - SYSTEM folder containing the HTML and XSL files used for Short Transaction error handling.
    - NEEEXST folder containing example web page templates for the example Natural Short Transaction objects.
  - NEEATM folder containing the Attach Manager executable and initialization file.
  - NEEC folder containing the web server interface program and initialization file.

## Installing WebStar under OS/390 and VSE/ESA

---

This section describes the installation required for a mainframe running under either OS/390 or VSE/ESA operating systems.

For OS/390, this will install the WebStar development function and the WebStar run time components.

For VSE/ESA, this will install the WebStar run time components only.

The installation of the Natural Engineer add-on component WebStar on mainframe platforms is performed by installation jobs. These jobs are either adapted “manually” or generated by SYSTEM MAINTENANCE AID (SMA).

For each step of the installation procedure described below, the job number of a job performing the respective task is indicated. This job number refers to an installation job generated by SMA. If you are not using SMA, a sample installation job of the same number is provided in the job library on the installation tape; you must adapt this example job to your requirements. Please note that the job numbers on the tape are preceded by a product code (for example, NWSI061).

### Prerequisites

The following products must be installed before you install Natural Engineer WebStar:

- Natural version 3.1.4 or above.
- Adabas version 7.1.2 or above.
- Natural Engineer version 4.4.2.

*Note: The version of Natural Engineer WebStar that is being installed must be identical to the version of Natural Engineer.*

### Using System Maintenance Aid

If you are using Software AG’s SYSTEM MAINTENANCE AID (SMA) for the installation process, please note the following before generating jobs:

1. Load the SMA table data as described in the SYSTEM MAINTENANCE AID manual (if you have not already done so).
2. Set NWS442 in the list of available products for your environment to “TO BE INSTALLED”.
3. Set the following SMA parameters specific to Natural Engineer WebStar:  
 In group OPTION:  
 NWS-FIRST-INSTALL = Y (for first-time installation of Natural Engineer WebStar)  
 NWS-FIRST-INSTALL = N (for migration installation of Natural Engineer WebStar)

## Tape Contents for OS/390

The installation tape contains the data sets listed in the table below. The sequence of the data sets is shown in the Report of Tape Creation that accompanies the installation tape. The notation vrs in Data Set Name represents the version number, release level and SM level of the product.

<b>Data Set Name</b>	<b>Contents</b>
NWSvrs.EXPL	Example application and Short Transaction examples.
NWSvrs.INPL	Natural Engineer WebStar system programs.
NWSvrs.IRES	Supplied Impact Search Criteria.
NWSvrs.JOBS	Job library with sample JCL.
NWSvrs.LOAD	Load library containing members NEEWEB and HAANUXIT.
NWSvrs.S001	WebStar run time components for web-enabled applications used for SMARTS under OS/390.
NWSvrs.S002	WebStar run time error handling components used for SMARTS under OS/390.
NWSvrs.S003	WebStar run time script components used for SMARTS under OS/390.
NWSvrs.WEBS	WebStar run time components for IBM WebSphere under OS/390 Unix System Services.

## Copying the Tape Contents to Disk for OS/390

If you are not using SMA, follow the instructions described below.

This section explains how to:

- Copy data set COPY.JOB from tape to disk.
- Modify this data set to conform to local site standards.

The JCL in data set COPY.JOB is used to copy all data sets from tape to disk.

After all the data sets have been copied, you will need to perform the individual install procedures for each component.

### Step 1 - Copy data set COPY.JOB from tape to disk.

The data set COPY.JOB (label 2) contains the JCL to unload all other existing data sets from tape to disk. To unload COPY.JOB use the following sample JCL:

```
//NWSTAPE JOB SAG,CLASS=1,MSGCLASS=X
//*
//COPY EXEC PGM=IEBGENER
//SYSUT1 DD DSN=COPY.JOB,DISP=(OLD,PASS),UNIT=(CASS,,DEFER),
// VOL=(,RETAIN,SER=<Tnnnnn>),LABEL=(2,SL)
//SYSUT2 DD DSN=<hilev>.COPY.JOB,DISP=(NEW,CATLG,DELETE),
// UNIT=3390,VOL=SER=<vvvvvv>,SPACE=(TRK,(1,1),RLSE),
// DCB=*.SYSUT1
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//*
```

Where:

<hilev> is a valid high level qualifier.

<Tnnnnn> is the tape number.

<vvvvvv> is the desired volser.

**Step 2 - Modify COPY.JOB to conform to local site standards.**

There are three parameters that must be set before submitting the COPY.JOB:

- Set HILEV to a valid high level qualifier.
- Set LOCATION to a storage location.
- Set EXPDT to a valid expiration date.

**Step 3 - Submit COPY.JOB**

Submit COPY.JOB to unload all other data sets from the tape to your disk.

## Installation Steps for OS/390

### 1. Load Natural Engineer WebStar System Programs

**SMA Reference:** Job I061, Step 7000

- ❑ The Natural Engineer WebStar system programs (SYSNEEW and NEEATM) are contained in the data set NWSvrs.INPL. Load them to your Natural FNAT and FUSER system files using the Natural utility INPL.

### 2. Load Example Application and Short Transaction Examples

**SMA Reference:** Job I061, Step 7002

- ❑ The example application (NEEEXWB) and Short Transaction examples (NEEEXST) are contained in the data set NWSvrs.EXPL. Load them to your Natural FUSER system files using the Natural utility INPL.

### 3. Load Natural Engineer WebStar Sample Impact Criteria

**SMA Reference:** JOB I500, Step 7020

- ❑ The Natural Engineer WebStar Impact criteria WEBSTAR are contained in the data set NWSvrs.IRES. Load them using member LOADIRES from the NWSvrs.JOBS data set.

## Tape Contents for VSE/ESA

The installation tape contains the data sets listed in the table below. The sequence of the data sets is shown in the Report of Tape Creation that accompanies the installation tape. The notation *vrs* in Data Set Name represents the version number, release level and SM level of the product.

<b>Data Set Name</b>	<b>Contents</b>
NWS <i>vrs</i> .EXPL	Example application and Short Transaction examples.
NWS <i>vrs</i> .INPL	Natural Engineer WebStar system programs.

## Copying the Tape Contents to Disk for VSE/ESA

If you are not using SMA, follow the instructions described below.

This section explains how to:

- Copy data set COPYTAPE.JOB from tape to disk.
- Modify this data set to conform to local site standards.

The JCL in data set COPYTAPE.JOB is used to copy all data sets from tape to disk.

After all the data sets have been copied, you will need to perform the individual install procedures for each component.

### **Step 1 - Copy data set COPYTAPE.JOB from tape to disk.**

The data set COPYTAPE.JOB (file 5) contains the JCL to unload all other existing data sets from tape to disk. To unload COPYTAPE.JOB use the following sample JCL:

```
* $$ JOB JNM=LIBRCAT,CLASS=0,
* $$ DISP=D,LDEST=(*,UID),SYSID=1
* $$ LST CLASS=A,DISP=D
// JOB LIBRCAT
* *****
*       CATALOG COPYTAPE.JOB TO LIBRARY
* *****
// ASSGN SYS004,NNN
// MTC REW,SYS004
// MTC FSF,SYS004,4
// ASSGN SYSIPT,SYS004
// TLBL IJSYSIN,'COPYTAPE.JOB'
// EXEC LIBR,PARM='MSHP; ACC S=lib.sublib'
// MTC REW,SYS004
// ASSGN SYSIPT,FEC
/*
/&
* $$ EOJ
```

Where:

NNN is the tape address.

lib.sublib is the catalog library name.

### Step 2 - Modify COPYTAPE.JOB to conform to local site standards.

Modify COPYTAPE.JOB to conform to local site standards and complete the disk space parameters before submitting the COPYTAPE.JOB.

### Step 3 - Submit COPYTAPE.JOB

Submit COPYTAPE.JOB to unload all other data sets from the tape to your disk.

## 2

### Natural Engineer WebStar

## Installation Steps for VSE/ESA

### 1. Load Natural Engineer WebStar System Programs

**SMA Reference:** Job I061, Step 7000

- The Natural Engineer WebStar system programs (SYSNEEW and NEEATM) are contained in the data set *NWSvrs.INPL*. Load them to your Natural FNAT and FUSER system files using the Natural utility INPL.

### 2. Load Example Application and Short Transaction Examples

**SMA Reference:** Job I061, Step 7002

- The example application (NEEEXWB) and Short Transaction examples (NEEEXST) are contained in the data set *NWSvrs.EXPL*. Load them to your Natural FUSER system files using the Natural utility INPL.

## Installing WebStar under Solaris 8

---

This section describes the installation required for a Unix computer running under the Sun Solaris 8 operating system.

This will install the WebStar run time components only.

*Note: The documentation relates to the Unix environment where case sensitivity is important. Any described input settings should be treated as read.*

### Prerequisites

The following products must be installed before you install Natural Engineer WebStar:

- Sun Solaris 8.
- Natural version 5.1.1 Patch Level 5 or above.
- Adabas version 3.1.1.56 or above.

### Installation Process

The installation process under Sun Solaris 8 requires the relevant files and folders to be copied from the Natural Engineer product CD.

All the WebStar run time function components are located in the following directory path:

```
/Unix/Sun/Nws/v442
```

The following summarizes each of the main folders within this directory path:

- /Unix/Sun/Nws/v442/sysobjh  
Contains the following Natural libraries:
  - NEEEXST - example Natural Short Transaction objects.
  - SYSNEEW - the WebStar run time objects utilized by the modified application when executed via the Internet.
- /Unix/Sun/Nws/v442/web/nee

## 2

### Natural Engineer WebStar

Contains the base WebStar HTML files and style sheets used at run time for web enabled applications. Also contains the SYSTEM folder containing the HTML and XSL files used for Short Transaction error handling.

- /Unix/Sun/Nws/v442/web/neeatm

Contains the Attach Manager executable and initialization file.

- /Unix/Sun/Nws/v442/web/neeec

Contains the web server interface program and initialization file.

- /Unix/Sun/Nws/v442/web/neeexst

Contains example web page templates for the example Natural Short Transaction objects.

### Load Natural Libraries

Use the SYSOBJH facility within Natural to load the files supplied in:

```
/Unix/Sun/Nws/v442/sysobjh
```

### Create WebStar Directory Structure

#### Default Directories

Default directories need to be created under the directory being used for WebStar run time. For example:

```
/sag/nws
```

#### Scripts

The contents of the folder located in /Unix/Sun/Nws/v442/web/neeec need to be added to the '/scripts' directory. For example:

```
/sag/nws/scripts
```

**Attach Manager (NEEATM)**

The contents of the folder located in /Unix/Sun/Nws/v442/web/neeatm need to be copied to the '/neeatm' directory. For example:

```
/sag/nws/neeatm
```

**Run Time Components.**

The contents of the folder (including sub-folders) located in /Unix/Sun/Nws/v442/web/nee need to be copied to the '/nee' directory. For example:

```
/sag/nws/nee
```

**Applications**

The contents of the folder located in /Unix/Sun/Nws/v442/web/neeexst need to be added to the '/nee' directory. For example:

```
/sag/nws/nee/neeexst
```

For any web enabled applications, create directories as sub-directories under the '/nee' directory. For example:

```
/sag/nws/nee/neeexwb
```

```
/sag/nws/nee/payroll
```

```
/sag/nws/nee/test
```

## Installing WebStar for SMARTS under OS/390

---

This section describes the installation required for a mainframe running under the OS/390 operating system and using the SMARTS HTTP server component.

This will install the WebStar development function and the WebStar run time components.

The installation of the Natural Engineer add-on component WebStar on mainframe platforms is performed by installation jobs. These jobs are either adapted “manually” or generated by SYSTEM MAINTENANCE AID (SMA).

For each step of the installation procedure described below, the job number of a job performing the respective task is indicated. This job number refers to an installation job generated by SMA. If you are not using SMA, a sample installation job of the same number is provided in the job library on the installation tape; you must adapt this example job to your requirements. Please note that the job numbers on the tape are preceded by a product code (for example, NWSI061).

### Prerequisites

The following products must be installed before you install Natural Engineer WebStar:

- Natural version 3.1.4 or above.
- Adabas version 7.1 or above.
- Natural Engineer version 4.4.2.

*Note: The version of Natural Engineer WebStar that is being installed must be identical to the version of Natural Engineer.*

### Using System Maintenance Aid

If you are using Software AG’s SYSTEM MAINTENANCE AID (SMA) for the installation process, please note the following before generating jobs:

1. Load the SMA table data as described in the SYSTEM MAINTENANCE AID manual (if you have not already done so).

2. Set NWS442 in the list of available products for your environment to “TO BE INSTALLED”.
3. Set the following SMA parameters specific to Natural Engineer WebStar:  
In group OPTION:  
NWS-FIRST-INSTALL = Y (for first-time installation of Natural Engineer WebStar)  
NWS-FIRST-INSTALL = N (for migration installation of Natural Engineer WebStar)

## Tape Contents for SMARTS under OS/390

The installation tape contains the data sets listed in the table below. The sequence of the data sets is shown in the Report of Tape Creation that accompanies the installation tape. The notation *vrs* in Data Set Name represents the version number, release level and SM level of the product.

<b>Data Set Name</b>	<b>Contents</b>
NWS <i>vrs</i> .EXPL	Example application and Short Transaction examples.
NWS <i>vrs</i> .INPL	Natural Engineer WebStar system programs.
NWS <i>vrs</i> .IRES	Supplied Impact Search Criteria.
NWS <i>vrs</i> .JOBS	Job library with sample JCL.
NWS <i>vrs</i> .LOAD	Load library containing members NEEWEB and HAANUXIT.
NWS <i>vrs</i> .S001	WebStar run time components for web-enabled applications used for SMARTS under OS/390.
NWS <i>vrs</i> .S002	WebStar run time error handling components used for SMARTS under OS/390.
NWS <i>vrs</i> .S003	WebStar run time script components used for SMARTS under OS/390.
NWS <i>vrs</i> .WEBS	WebStar run time components for IBM WebSphere under OS/390 Unix System Services.

## Copying the Tape Contents to Disk for SMARTS under OS/390

If you are not using SMA, follow the instructions described below.

This section explains how to:

- Copy data set COPY.JOB from tape to disk.
- Modify this data set to conform to local site standards.

The JCL in data set COPY.JOB is used to copy all data sets from tape to disk.

After all the data sets have been copied, you will need to perform the individual install procedures for each component.

### Step 1 - Copy data set COPY.JOB from tape to disk.

The data set COPY.JOB (label 2) contains the JCL to unload all other existing data sets from tape to disk. To unload COPY.JOB use the following sample JCL:

```
//NWSTAPE JOB SAG,CLASS=1,MSGCLASS=X
//*
//COPY EXEC PGM=IEBGENER
//SYSUT1 DD DSN=COPY.JOB,DISP=(OLD,PASS),UNIT=(CASS,,DEFER),
// VOL=(,RETAIN,SER=<Tnnnnn>),LABEL=(2,SI)
//SYSUT2 DD DSN=<hilev>.COPY.JOB,DISP=(NEW,CATLG,DELETE),
// UNIT=3390,VOL=SER=<vvvvvv>,SPACE=(TRK,(1,1),RLSE),
// DCB=*.SYSUT1
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//*
```

Where:

<hilev> is a valid high level qualifier.

<Tnnnnn> is the tape number.

<vvvvvv> is the desired volser.

**Step 2 - Modify COPY.JOB to conform to local site standards.**

There are three parameters that must be set before submitting the COPY.JOB:

- Set HILEV to a valid high level qualifier.
- Set LOCATION to a storage location.
- Set EXPDT to a valid expiration date.

**Step 3 - Submit COPY.JOB**

Submit COPY.JOB to unload all other data sets from the tape to your disk.

## Installation Steps for SMARTS under OS/390

### 1. Load Natural Engineer WebStar System Programs

**SMA Reference:** Job I061, Step 7000

- ❑ The Natural Engineer WebStar system programs (SYSNEEW and NEEATM) are contained in the data set NWSvrs.INPL. Load them to your Natural FNAT and FUSER system files using the Natural utility INPL.

### 2. Load Example Application and Short Transaction Examples

**SMA Reference:** Job I061, Step 7002

- ❑ The example application (NEEEXWB) and Short Transaction examples (NEEEXST) are contained in the data set NWSvrs.EXPL. Load them to your Natural FUSER system files using the Natural utility INPL.

### 3. Load Natural Engineer WebStar Sample Impact Criteria

**SMA Reference:** JOB I500, Step 7020

- ❑ The Natural Engineer WebStar Impact criteria WEBSTAR are contained in the data set NWSvrs.IRES. Load them using member LOADIRES from the NWSvrs.JOBS data set.



# CONFIGURATION

## Chapter Overview

---

This chapter explains how to configure the Natural Engineer add-on component WebStar. The configurations apply to the run time components required to successfully execute a web enabled application via the Internet.

The following topics are covered:

- [Configuration Options](#)
- [Web Server](#)
- [EntireX Broker](#)
- [Natural](#)

## Configuration Options

---

### Infrastructure

To successfully execute a web enabled Natural application via the Internet requires the following infrastructure:

#### Web Browser

This will be the user interface for executing the web enabled Natural application.

#### Web Server

Used to communicate between the web browser and the Natural application.

#### EntireX Broker

Provides the communication mechanism between the web server and a Natural session.

*Note 1: If running using IBM WebSphere HTTP server under OS/390 Unix System Services then no EntireX Broker is required. Instead a broker stub needs to be provided to provide the communication path between WebSphere and Natural.*

*Note 2: If running SMARTS then there is no need to use EntireX as it provides its own communication mechanism.*

#### Natural

The WebStar run time component utilizes Natural server sessions to execute the Natural application code. The WebStar Attach Manager program NEEATM controls this process and provides the communication between Natural and EntireX Broker.

*Note: If you are using the SMARTS HTTP server, then you have the option of using its own attach manager NATCGI.*

### Product and Platform Support

The following table illustrates the various products and platforms supported by the Natural Engineer add on component WebStar.

	Windows	Solaris 8	OS/390	VSE/ ESA	OS/390 USS <sup>1</sup>
<b>Web Browsers</b>					
Internet Explorer version 5 or above	✓ <sup>2</sup>				
Netscape version 6 or above	✓				
Mozilla version 5 or above	✓				
<b>Web Servers</b>					
Microsoft IIS version 4 or above	✓ <sup>3</sup>				
Apache HTTP server version 1.3 or above	✓ <sup>3</sup>	✓ <sup>4</sup>			
IBM WebSphere version 1.3.19 or above	✓ <sup>3</sup>				
Sun iPlanet version 6.02	✓ <sup>3</sup>	✓ <sup>4</sup>			
SMARTS HTTP Server version 2.6.1.6 or above			✓		
IBM WebSphere HTTP Server version 5.3 or above					✓
<b>Message Orientated Middleware</b>					
EntireX 5.3.2				✓	
EntireX Communicator 6.2.2 or above	✓	✓	✓		
<b>TP Monitor</b>					
Com-Plete version 6.1			✓	✓	
CICS			✓	✓	

*Note: Where more than one product per platform is indicated, only one of the products is required.*

<sup>1</sup> Unix System Services.

<sup>2</sup> It is recommended that Internet Explorer version 6 is used.

<sup>3</sup> To enable server side XSL and XML data merging, Internet Explorer version 6 needs to be installed on the web server. This applies to Windows environments only.

<sup>4</sup> It is recommended that Internet Explorer version 6 is installed on the client side to provide client side XSL and XML data merging.

## Web Server

---

This section describes all the configuration requirements for Web Servers.

### HTML/XSL Folders

#### Windows

##### Microsoft IIS

Copy all the contents (including sub-folders) of the NEE folder located in X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\WEB folder (where X: is the directory on which Natural Engineer was installed) to Y:\INETPUB\WWWROOT\NEE folder (where Y: is the location of the web server).

##### Apache

Copy all the contents (including sub-folders) of the NEE folder located in X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\WEB folder (where X: is the directory on which Natural Engineer was installed) to Y:\PROGRAM FILES\APACHE GROUP\APACHE\HTDOCS\NEE folder (where Y: is the location of the web server).

##### IBM WebSphere

Copy all the contents (including sub-folders) of the NEE folder located in X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\WEB folder (where X: is the directory on which Natural Engineer was installed) to Y:\IBM HTTP SERVER\HTDOCS\NEE folder (where Y: is the location of the web server).

##### Sun iPlanet

Copy all the contents (including sub-folders) of the NEE folder located in X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\WEB folder (where X: is the directory on which Natural Engineer was installed) to Y:\IPLANET\SERVERS\DOCS\NEE folder (where Y: is the location of the web server).

## OS/390 Unix System Services

The IBM WebSphere product contains the HTTP server, which can be used to run the WebStar run time function under OS/390 Unix System Services environments.

This section is based on the HTTP server version 5.3 running under OS/390 version 2 release 10.

*Note: The documentation relates to the Unix environment where case sensitivity is important. Any described input settings should be treated as read.*

### IBM WebSphere

The WebStar run time function components need to be added as a '/nee' directory under the pub directory. For example:

```
/usr/lpp/internet/server-root/pub/nee
```

## Web Server Interface Program NEEC.EXE

The web server interface program NEEC.EXE, is used to control the communication between the web server and EntireX Broker.

The execution parameters for NEEC.EXE, are defined in an initialization file NEEC.INI.

The NEEC.INI file contains the parameter settings required to map to the correct EntireX Broker service.

There are two sets of configuration required:

1. Copy the contents of the installed NEEC folder to the web server scripts folder.
2. Configure the interface program initialization file NEEC.INI.

*Note: All the aliases described are case sensitive and should be input as per the documentation.*

# 3

## Natural Engineer WebStar

### NEEC Folder Contents

#### Windows or Solaris 8

##### Microsoft IIS

Copy all the contents of the NEEC folder located in X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\WEB folder (where X: is the directory on which Natural Engineer was installed) to Y:\INETPUB\SCRIPTS folder (where Y: is the location of the web server).

##### Apache

Copy all the contents of the NEEC folder located in X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\WEB folder (where X: is the directory on which Natural Engineer was installed) to Y:\PROGRAM FILES\APACHE GROUP\APACHE\CGI-BIN folder (where Y: is the location of the web server).

##### IBM WebSphere

Copy all the contents of the NEEC folder located in X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\WEB folder (where X: is the directory on which Natural Engineer was installed) to Y:\IBM HTTP SERVER\CGI-BIN folder (where Y: is the location of the web server).

##### Sun iPlanet

Copy all the contents of the NEEC folder located in X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\WEB folder (where X: is the directory on which Natural Engineer was installed) to Y:\IPLANET\SERVERS\SCRIPTS folder (where Y: is the location of the web server).

#### OS/390 Unix System Services

The IBM WebSphere product contains the HTTP server, which can be used to run the WebStar run time function under OS/390 Unix System Services environments.

This section is based on the HTTP server version 5.3 running under OS/390 version 2 release 10.

*Note: The documentation relates to the Unix environment where case sensitivity is important. Any described input settings should be treated as read.*

**IBM WebSphere**

The WebStar script objects need to be added to the '/cgi-bin' directory. For example:

```
/usr/lpp/internet/server-root/cgi-bin
```

**NEEC.EXE Initialization File NEEC.INI****Windows**

The NEEC.EXE initialization file NEEC.INI is located in the same 'scripts' folder as NEEC.EXE. This can be located based on the actions taken in [NEEC Folder Contents](#).

<b>Group header / Parameter</b>	<b>Description</b>
<b>NEE_USER_ID=</b>	Default value = NEEWEB Provides a user identity when executing Short Transactions.
<b>NEE_PASSWORD=</b>	Default value = NEEWEB Provides a password when executing Short Transactions.
<b>NEE_TOKEN=</b>	Default value = NEEWEB Allows the application to reconnect at a different terminal without losing the existing conversation. Mainly used for Short Transactions.
<b>[DEFAULT]</b>	
<b>ETB_ID_NAME=</b>	Default value = ETB255 This is the Broker identity to be used. Mainly used for Short Transactions.
<b>ETB_CLASS_NAME=</b>	Default value = NEE This is the CLASS name specified in the EntireX Broker Attributes File under the server-specific section. Mainly used for Short Transactions.
<b>ETB_SERVER_NAME=</b>	Default value = NEEWEB This is the SERVER name specified in the EntireX Broker Attributes File under the server-specific section. Mainly used for Short Transactions.

## 3

## Natural Engineer WebStar

Group header / Parameter	Description
<b>ETB_SERVICE_NAME=</b>	<p>Default value = NEEWEB</p> <p>This is the SERVICE name specified in the EntireX Broker Attributes File under the server-specific section.</p> <p>Mainly used for Short Transactions.</p>
<b>ETB_TIME_OUT=</b>	<p>Default value = 60S</p> <p>Controls the elapsed time between browser pages. The values are input using EntireX Broker conventions.</p> <p>Possible values are:</p> <p><b>N</b> Value is in seconds, for example 60 is 60 seconds.  <b>Ns</b> Value is in seconds, for example 60s is 60 seconds.  <b>Nm</b> Value is in minutes, for example 60m is 60 minutes.  <b>Nh</b> Value is in hours, for example 60h is 60 hours.</p>
[NEE]	
<b>HTML_TEMPLATE_DIR=</b>	<p>Default value = Y:\INETPUB\WWWROOT\NEE (where Y: is the location of the web server.)</p> <p>The location on the web server where the XSL files are located.</p> <p>Possible values are:</p> <p><b>Microsoft IIS</b> Y:\INETPUB\WWWROOT\NEE  <b>Apache</b> Y:\PROGRAM FILES\APACHE GROUP\APACHE\HTDOCS\NEE  <b>IBM WebSphere</b> Y:\IBM HTTP SERVER\HTDOCS\NEE  <b>Sun iPlanet</b> Y:\IPLANET\SERVERS\DOCS\NEE</p> <p><i>Note: Y: is the location of the web server.</i></p>
<b>XSL_URL=</b>	<p>Default value = /nee</p> <p>Determines the high level folder name where the application XSL files are located.</p> <p><i>Note: The XSL files can be located in a Tamino database. This would be specified as:</i></p> <p><i>XSL_URL=http://IP address/Tamino/xsl-db-name/nee.</i></p>
[NEEATM]	
<b>APP_FILE=</b>	<p>Default value = X:\PROGRAM FILES\SOFTWARE AG \NATURAL ENGINEER\4.4.2\WEB\NEEATM\NEEATM.INI (where X: is the directory where Natural Engineer has been installed.)</p> <p>The location of the Attach Manager (NEEATM) initialization files.</p>

Group header / Parameter	Description
[MISC]	
<b>TEMP_DIR=</b>	Default value = C:\TEMP The location where temporary files are placed during processing.
<b>XML_BROWSER=</b>	Default value = IE6 Used to determine which client web browsers support the XSL/XML merge process. Possible value is: IE6 (Internet Explorer 6)
<b>XML_ENCODING=</b>	Default value = ISO-8859-1 Defines the character set to be used to translate any special characters. For example: <b>XML_ENCODING=windows-1251</b> Will use the Cyrillic character set.
<b>ERROR_TEMPLATE=</b>	Default value = NEEERROR.HTML Used to assist error tracing within the web environment.
<i>Note: The following parameters are only used for diagnostic purposes. They are only required if requested by Support.</i>	
<b>ERROR_LOG_FILE=</b>	Default value = NEEC.LOG Used to assist error tracing within the web environment.
<b>TRACE_FILE=</b>	Default value = NEEC.HTML Used to assist error tracing within the web environment.
<b>DATA_DIR=</b>	Default value = C:\TEMP Location where all diagnostic log files are placed.

## 3

### Natural Engineer WebStar

#### OS/390 Unix System Services

The IBM WebSphere product contains the HTTP server, which can be used to run the WebStar run time function under OS/390 Unix System Services environments.

This section is based on the HTTP server version 5.3 running under OS/390 version 2 release 10.

*Note: The documentation relates to the Unix environment where case sensitivity is important. Any described input settings should be treated as read.*

The WebStar web interface program initialization file NEEC.INI needs to have the following parameter settings made:

```
HTML_TEMPLATE_DIR=/usr/lpp/internet/server_root/pub/nee  
APP_FILE=/usr/lpp/internet/server_root/pub/nee/neeatm.ini  
TEMP_DIR=/usr/lpp/internet/server_root/pub/nee/temp
```

*Note: The '/temp' directory must be created before using the WebStar run time function and it must be set up with WRITE access. The '/temp' directory is used to place all temporary files during processing.*

#### Solaris 8

This section describes the configuration required to run the WebStar run time function under Sun Solaris 8 operating environments.

The configuration information is based on the following web servers:

- Apache version 1.3.19
- iPlanet version 6.0.2

*Note: The documentation relates to the Unix environment where case sensitivity is important. Any described input settings should be treated as read.*

The WebStar web interface program initialization file NEEC.INI (located in the 'sag/nws/scripts' directory) needs to have the following parameter settings made:

```
HTML_TEMPLATE_DIR=/sag/nws/nee  
APP_FILE=/sag/nws/neeatm/neeatm.ini  
TEMP_DIR=/sag/nws/nee/temp
```

*Note: The '/temp' directory must be created before using the WebStar run time function and it must be set up with WRITE access. The '/temp' directory is used to place all temporary files during processing.*

### SMARTS under OS/390

The Software AG Multiple Architecture Runtime System (SMARTS) environment contains the HTTP server component, which can be used to run the WebStar run time function under OS/390 environments.

This section is based on the SMARTS HTTP server version 2.6.1.6 running under OS/390 version 2 release 10 using Com-Plète version 6.1.

The WebStar web interface program initialization file NEEC.INI needs to have the following parameter settings made:

```
HTML_TEMPLATE_DIR=FILE://NEE  
APP_FILE= FILE://NEE/ATMINI  
TEMP_DIR= FILE://NEE/TEMP
```

*Note 1: The '/TEMP' PDS data set must be created before using the WebStar run time function and it must be set up with WRITE access. The '/TEMP' PDS data set is used to place all temporary files during processing.*

*Note 2: Under SMARTS, the NEEC.INI file is the NEECINI member within the NEE PDS data set.*

*Note 3: The HLQ that is applied will need to follow local site standards. If NEE cannot be used as the HLQ, then it should immediately follow the site standard HLQ, for example SITEHLQ.NEE.*

## Attach Manager NEEATM

The WebStar run time component contains the Attach Manager program NEEATM.EXE, which is used to initiate and maintain individual Natural sessions when executing a web enabled Natural application via the Internet using Windows or Solaris 8 operating systems.

The execution parameters for the Attach Manager program NEEATM.EXE are defined in an initialization file NEEATM.INI.

The NEEATM.INI file performs two roles:

1. Provides a list of Applications to log on to.
2. Provides the link between the Natural session and EntireX Broker.

## Windows

The NEEATM.INI file contains the following parameters:

NEEATM.INI Parameter	Description
<b>BROKER-ID=</b>	Default value = ETB255 The default Broker service identity.
<b>APP-DEF</b>	This is a section heading tag, which indicates that the following parameters form the application definition.
<b>APP-TITLE=</b>	This is the name of the application that is displayed on the WebStar logon page (LOGON.HTML). If an entry has no title, then it will not appear in the list of applications.
<b>APP-BROKER-ID=</b>	This is the application Broker service identity to be used during execution. Each application can execute using different identities.
<b>APP-SERVER-CLASS=</b>	This is the CLASS name specified in the EntireX Broker Attributes File under the server-specific section.
<b>APP-SERVER-NAME=</b>	This is the SERVER name specified in the EntireX Broker Attributes File under the server-specific section.
<b>APP-SERVICE=</b>	This is the SERVICE name specified in the EntireX Broker Attributes File under the server-specific section.

NEEATM.INI Parameter	Description
<b>APP-EXEC=</b>	<p>Default value = X:\PROGRAM FILES\SOFTWARE AG \NATURAL\5.1.1\BIN\NATURAL.EXE (where X: is the directory where Natural has been installed.)</p> <p>Defines the Natural start up program for the Natural session.</p> <p><i>Note: This parameter setting is not required if executing the application using a mainframe Natural session.</i></p>
<b>APP-PARM=</b>	<p>This is the Natural parameter information used to invoke the Natural application. There are two formats that can be used:</p> <p>1. For WebStar applications:            PARM=PPPPPPPP STACK=            (LOGON LLLLLLLL user-id password;WEB:SSSSSSSS)</p> <p>Where:</p> <ul style="list-style-type: none"> <li><b>PPPPPPPP</b> The name of the Natural Parameter module, for example NEEEXWB.</li> <li><b>LLLLLLLL</b> The name of the Natural library to be used at Natural start up.</li> <li><b>user-id</b> A logon user identity.</li> <li><b>password</b> A logon password.</li> <li><b>WEB:</b> The WebStar run time application control program.</li> <li><b>SSSSSSSS</b> The Natural application start up program, for example WEBP1.</li> </ul> <p>2. For Short Transactions:            PARM=PPPPPPPP STACK=            (LOGON LLLLLLLL user-id password;NEEWEB)</p> <p>Where:</p> <ul style="list-style-type: none"> <li><b>PPPPPPPP</b> The name of the Natural Parameter module, for example NEEEXST.</li> <li><b>LLLLLLLL</b> The name of the Natural library to be used at Natural start up.</li> <li><b>user-id</b> A logon user identity.</li> <li><b>password</b> A logon password.</li> <li><b>NEEWEB</b> The WebStar run time Short Transaction control program.</li> </ul> <p><i>Note: a logon user identity and password are only required if the run time environment utilizes Natural Security.</i></p>

### 3

#### Natural Engineer WebStar

*Note: Both the NEEATM.EXE and NEEATM.INI objects can be located in X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\WEB\NEEATM folder (where X: is the directory on which Natural Engineer was installed).*

#### Solaris 8

The WebStar attach manager program initialization file NEEATM.INI (located in the 'sag/nws/neeatm' directory) needs to have the following parameter settings applied:

Location of Natural executable controlled by the APP\_EXEC parameter:

```
APP_EXEC=$NATDIR/$NATVERS/bin/natural
```

Natural parameter information controlled by the APP\_PARM parameter, requires an '&' at the end of the string and the STACK information bound by single quotes. For example:

```
APP_PARM=PARAM=NEEEXWBX 'STACK=(LOGON NEEEXWBX;WEB:WEBP1) ' &
```

*Note 1: All other parameters are as per Windows operating system.*

*Note 2: The documentation relates to the Unix environment where case sensitivity is important. Any described input settings should be treated as read.*

#### OS/390 using Com-Plete or CICS

The NEEATM.INI file is only used to provide a list of applications to log on to, using the "APP-TITLE=" parameter.

*Note: The WebStar Attach Manager program NEEATM is controlled by Natural subroutines NEEATA-S and NEEATI-S. For more information on configuration settings for the OS/390 environment using Com-Plete or CICS, refer to the configuration section [Natural](#).*

## Broker Stub

### Windows

If the WebStar run time function is to be run with the web server and EntireX Broker on separate machines, then you will need to install a mini runtime installation of EntireX. This will install the relevant Broker stubs required.

To install, insert the EntireX CD into the web server machine and switch to the following directory: X:\WINDOWS\EXX\PROGRAM FILES\SOFTWARE AG\ENTIREX\ETC folder, where X: is the location of the CD drive. Then invoke the executable file EntireXMiniRuntime.EXE.

### OS/390 under Unix System Services

If the WebStar run time function is to be run using the IBM WebSphere HTTP server under OS/390 Unix System Services environment, then the Broker stub 'broker.dll' needs to be used.

The Broker stub 'broker.dll' provides the mechanism to communicate between Natural and the IBM WebSphere HTTP server.

*Note: The documentation relates to the Unix environment where case sensitivity is important. Any described input settings should be treated as read.*

The Broker stub can be installed as follows:

1. Create a Unix System Services directory:  
`/sagprod`
2. FTP the following file from the EntireX product CD, using 'binary' mode to the Unix System Services directory '/sagprod':  
`/Unix/Os390/Exx/v611/A32b/Exar.cpz`
3. Logon to Unix Systems Services and position to the directory '/sagprod' and issue the following command:  
`cpio -icBvmd < Exar.cpz`
4. The Broker stub 'broker.dll' is now installed in the following Unix System Services directory:  
`/sagprod/exx/v611/lib/broker.dll`

## 3

### Natural Engineer WebStar

*Note: If EntireX version 5.3.1 is installed, then the Broker stub is installed by installing the Developer Kit 'Exx531.cpio'. For more information refer to the EntireX version 5.3.1 documentation.*

## Web Server Configuration Settings

### Windows

#### Microsoft IIS

If you are using the standard web server interface program NEEC.EXE, then no further configuration is required.

If you wish to make use of ISAPI functionality on an IIS web server, then you need to change the scripts mapping within the template member TEMPM. This needs to have the following FORM NAME setting:

```
<form name="logon" method="post" action="/scripts/neeisapi.dll">
```

#### Apache

The following entry needs to be added to the HTTPD.CONF file:

- ScriptAlias /scripts/ "Y:/Program Files/Apache Group/Apache/cgi-bin/"

(where Y: is the location of the web server)

#### Microsoft IIS and Apache on the same web server

If both Microsoft IIS and Apache are installed on the web server, to avoid duplication of scripts and HTML/XSL files, the Apache HTTPD.CONF file can be amended to point to the Microsoft IIS INETPUB folders.

Example for scripts:

- ScriptAlias /scripts/ "Y:/Inetpub/scripts/"

(where Y: is the location of the web server)

Example for HTML/XSL files:

- Alias /nee/ "Y:/Inetpub/wwwroot/nee/"

(where Y: is the location of the web server)

### IBM WebSphere

The following entry needs to be added to the HTTPD.CONF file:

- ScriptAlias /scripts/ "Y:/IBM HTTP Server/cgi-bin/"

(where Y: is the location of the web server)

### Microsoft IIS and IBM WebSphere on the same web server

If both Microsoft IIS and IBM WebSphere are installed on the web server, to avoid duplication of scripts and HTML/XSL files, the IBM WebSphere HTTP Server HTTPD.CONF file can be amended to point to the Microsoft IIS INETPUB folders.

Example for scripts:

- ScriptAlias /scripts/ "Y:/Inetpub/scripts/"

(where Y: is the location of the web server)

Example for HTML/XSL files:

- Alias /nee/ "Y:/Inetpub/wwwroot/nee/"

(where Y: is the location of the web server)

### Sun iPlanet

The following entries need to be added using the Administration Server options Class Manager → Programs → CGI Directory:

- Add prefix: /scripts
- CGI Directory: Y:/IPLANET/SERVERS/SCRIPTS

(where Y: is the location of the web server)

### 3

#### Natural Engineer WebStar

##### Microsoft IIS and Sun iPlanet on the same web server

If both Microsoft IIS and Sun iPlanet are installed on the web server, to avoid duplication of scripts and HTML/XSL files, the Administration Server can be used to point to the Microsoft IIS INETPUB folders.

Example for scripts:

Administration Server options Class Manager → Programs → CGI Directory

- Add prefix: /scripts
- CGI Directory: Y:/INETPUB/SCRIPTS

(where Y: is the location of the web server)

Example for HTML/XSL files:

Administration Server options Class Manager → Content Management → Additional Document Directories:

- Add prefix: /nee
- CGI Directory: Y:/INETPUB/WWWROOT/NEE/

(where Y: is the location of the web server)

## OS/390 Unix System Services

The IBM WebSphere product contains the HTTP server, which can be used to run the WebStar run time function under OS/390 Unix System Services environments.

This section is based on the HTTP server version 5.3 running under OS/390 version 2 release 10.

*Note: The documentation relates to the Unix environment where case sensitivity is important. Any described input settings should be treated as read.*

### HTTP Server Configuration File Settings

The following entries need to be added to the HTTP server configuration file located in etc/httpd.conf.

#### 1. Exec Directive.

Map the script alias '/scripts/' to the cgi-bin. For example:

```
Exec    /scripts/*    /usr/lpp/internet/server-root/cgi-bin/*
```

#### 2. AddType Directive.

Set the MIME type / sub type for JavaScript, XML, XSL and CSS file suffixes. For example:

```
AddType .js    text/javascript    ebcDic    1.0 # JavaScript
AddType .xml    text/xml            ebcDic    1.0 # XML
AddType .xsl    text/xsl            ebcDic    1.0 # XSL
AddType .css    text/css            ebcDic    1.0 # CSS
```

*Note: Some or all of these settings may already be in place.*

#### 3. InheritEnv Directive.

Set the environment variable ETBLNK to be inherited by CGI to call the Broker ACI. For example:

```
InheritEnv  ETBLNK
```

*Note: This setting is not required if no other InheritEnv directives exist in the configuration file, as then ALL environment variables are inherited.*

# 3

## Natural Engineer WebStar

### HTTP Server Environment Variables

The environment variable ETBLNK needs to be added to the HTTP server environment variable file located in etc/httpd.envvars. For example:

```
ETBLNK=/SAGPROD/exx/v611/lib/broker.dll
```

### User Permissions

The correct access permissions need to be specified for the scripts and application templates.

#### 1. Scripts

Permission should be set as Read Execute. For example:

```
CHMOD 755 *
```

#### 2. Application templates

Permission should be set as Read Only. For example:

```
CHMOD 644 *
```

*Note: All application template names must be in lowercase.*

## Solaris 8

This section describes the configuration required to run the WebStar run time function under Sun Solaris 8 operating environments.

The configuration information is based on the following web servers:

- Apache version 1.3.19
- iPlanet version 6.0.2

*Note: The documentation relates to the Unix environment where case sensitivity is important. Any described input settings should be treated as read.*

## Apache

The following entries need to be added to the Apache Server configuration file 'httpd.conf'.

### 1. ScriptAlias Directive.

Map the script alias '/scripts' as containing CGI scripts. For example:

```
ScriptAlias /scripts/ "/sag/nws/scripts/"
```

### 2. Alias Directive.

Map the location of the WebStar run time components (HTML files, style sheets). For example:

```
Alias /nee/ "/sag/nws/nee/"
```

### 3. SetEnv Directive.

Add the environment variable ETBLNK used to provide the necessary link to EntireX Broker. For example:

```
SetEnv ETBLNK "/sag/exx/v611/lib/broker.so"
```

# 3

## Natural Engineer WebStar

### iPlanet

The following entries need to be added to the iPlanet Server configuration settings:

#### 1. CGI Directory.

Map the script alias '/scripts' as containing CGI scripts. This can be done using the Administration Server options Class Manager → Programs → CGI Directory. For example:

```
Add prefix: /scripts
```

```
CGI Directory: /sag/nws/scripts
```

#### 2. Content Management.

Map the location of the WebStar run time components (HTML files, style sheets). This can be done using the Administration Server options Class Manager → Content Management → Additional Document Directories. For example:

```
Add prefix: /nee
```

```
CGI Directory: /sag/nws/nee
```

#### 3. Environment Variable.

Add the environment variable ETBLNK used to provide the necessary link to EntireX Broker. This needs to be done manually in the 'magnus.conf' file. For example:

```
Init fn=init-cgi ETBLNK=/sag/exx/v611/lib/broker.so
```

### User Permissions

The correct access permissions need to be specified for the scripts and application templates.

#### 1. Scripts

Permission should be set as Read Execute. For example:

```
CHMOD 755 *
```

#### 2. Application templates

Permission should be set as Read Only. For example:

```
CHMOD 644 *
```

*Note: All application template names must be in lowercase.*

## SMARTS under OS/390

The Software AG Multiple Architecture Runtime System (SMARTS) environment contains the HTTP server component, which can be used to run the WebStar run time function under OS/390 environments.

This section is based on the SMARTS HTTP server version 2.6.1.6 running under OS/390 version 2 release 10 using Com-plete version 6.1.

### Default Directories

The SMARTS HTTP server makes use of PDS data sets to store the WebStar run time components as PDS members. The naming convention of the PDS members is restricted to 8 bytes, this means that any file extensions for HTML, XSL etc are ignored.

PDS data sets need to be created for each of the WebStar run time component directories.

Each PDS data set should have an Organization of '**PO**', a Record Format of '**VB**' and a Record Length of '**260**' specified. All other characteristics can be specified as per site standards.

The High Level Qualifier (HLQ) will be typically set to NEE.

*Note: The HLQ that is applied will need to follow local site standards. If NEE cannot be used as the HLQ, then it should immediately follow the site standard HLQ, for example SITEHLQ.NEE.*

The following examples are based on a HLQ of NEE.

### Scripts and Run Time Components

#### 1. Scripts

The SMARTS HTTP server does not have a '/scripts' directory. The WebStar script object comes supplied within a load PDS member, which is concatenated to COMPLIB in the Com-plete load JCL.

To execute the web server interface program NEEC, when using the SMARTS HTTP server, requires a change to the scripts mapping within the template member TEMPM. This needs to have the following FORM NAME setting:

```
<form name="logon" method="post" action="/cgi/nec">
```

# 3

## Natural Engineer WebStar

To execute the Natural CGI program, when using the SMARTS HTTP server, requires a change to the scripts mapping within the template member TEMPM. This needs to have the following FORM NAME setting:

```
<form name="logon" method="post" action="/natcgi/sysneew/neeapl">
```

### 2. Run Time Components

Create PDS data set NEE.

The WebStar run time function components used for web-enabled applications need to be added:

```
ATMINI  
GSLLOGO  
GSLLOGOB  
LOGOFF  
LOGON  
NEECINI  
NEE  
NEEERROR  
NEEVAL  
NEEWEB  
NEEWEBL  
TEMP  
TEMPL  
TEMPM  
TEMPT  
WEBSTAR
```

Create a PDS data set NEE.SYSTEM.

The WebStar run time function components used for error handling need to be added:

```
NATERR
```

*Note: Any other required Natural components should also be added to the NEE.SYSTEM PDS data set.*

### 3. Applications

Create a PDS data set for each application to be web-enabled. For example:

```
NEE.NEEEXWB  
NEE.PAYROLL  
NEE.TEST
```

**SMARTS HTTP Environment Variable**

An environment variable NEEHOME needs to be added to the SMARTS HTTP PDS member containing the environment variables being utilized. The format is:

```
NEEHOME=/nee
```

*Note 1: The HLQ that is applied will need to follow local site standards. If NEE cannot be used as the HLQ, then it should immediately follow the site standard HLQ, for example NEEHOME=/sitehlq/nee.*

*Note 2: All SMARTS HTTP environment variable names are upper case EBCDIC values.*

## EntireX Communicator Components

---

This section describes all the configuration requirements for the EntireX Broker, EntireX Broker Services and Broker Stub components of EntireX Communicator.

### EntireX Broker

EntireX Broker provides the communication mechanism required between the web server and the Natural session running the Natural application.

The EntireX Broker needs to be administered to support the Natural applications being executed. This requires a Broker Instance to be set up and its respective Broker Attribute File modified to define the Natural applications to be used.

### For All Platforms

#### Create a Broker Instance

A new Broker Service can be administered by using the EntireX Broker Administration screen and selecting the 'Add New Broker' option. A Broker name and Port number need to be specified, for example:

Broker ETB255 and Port number 1959.

#### Broker Attribute File

The Broker Attribute File contains the parameters that control the availability and characteristics of the clients and servers. It is divided into two sections, the Broker-specific and server-specific. For WebStar, modifications need to be applied to the server-specific section only.

The server-specific attribute section can be located by looking for the DEFAULTS=SERVICE parameter. The WebStar modifications require the addition of several service identities, consisting of the attributes CLASS, SERVER and SERVICE. These can be added after the sample definitions found in the Broker Attribute file:

```
CLASS = ACLASS, SERVER = ASERVER, SERVICE = ASERVICE  
CLASS = BCLASS, SERVER = BSERVER, SERVICE = BSERVICE  
CLASS = CCLASS, SERVER = CSERVER, SERVICE = CSERVICE
```

For the purpose of this section, the Natural application name will be NEEEXWBX.

There are three service identities that need to be added:

1. Service identity for the Natural application to identify the correct transaction when handling long transactions. This would be specified as:

```
CLASS = NEE, SERVER = NEEEXWBX, SERVICE = INPUT
```

*Note: Additional identities for each Natural application can be added, each time changing the SERVER= value to match the Natural application being referenced.*

2. Service identity for the Natural application to identify the correct transaction when handling short transactions. This would be specified as:

```
CLASS = NEE, SERVER = NEEWEB, SERVICE = NEEWEB
```

*Note: Only one of these identities needs to be defined within each Broker instance.*

3. Service identity to provide the ability to shutdown the attach manager. This would be specified as:

```
CLASS = NEE, SERVER = NEEATM, SERVICE = END
```

*Note: Only one of these entries needs to be defined within each Broker instance.*

To summarize, a Broker Attribute File utilized for the WebStar solution would contain the following definitions under the server-specific section:

```
CLASS = NEE, SERVER = NEEEXWBX, SERVICE = INPUT  
CLASS = NEE, SERVER = NEEWEB, SERVICE = NEEWEB  
CLASS = NEE, SERVER = NEEATM, SERVICE = END
```

Or, for multiple Natural applications:

```
CLASS = NEE, SERVER = NATAPP1, SERVICE = INPUT  
CLASS = NEE, SERVER = NATAPP2, SERVICE = INPUT  
CLASS = NEE, SERVER = NATAPP3, SERVICE = INPUT  
CLASS = NEE, SERVER = NEEWEB, SERVICE = NEEWEB  
CLASS = NEE, SERVER = NEEATM, SERVICE = END
```

# 3

## Natural Engineer WebStar

### Define Broker Host and Services

The Broker host needs to be defined so that the web server knows where the Broker resides. This is done using the Windows HOSTS file, which contains the mappings of IP addresses to host names. This file needs to be amended to map the IP address of the Broker.

The HOSTS file is typically located in the X:\WINNT\SYSTEM32\DRIVERS\ETC folder where X: is the directory where the Windows operating system is installed.

Example mapping for Broker ETB255:

```
127.0.0.1 ETB255
```

If the Broker utilizes a non-standard port (the standard Broker port is 1971), then the Windows SERVICES file needs to have a Broker service and port added.

The SERVICES file is typically located in the X:\WINNT\SYSTEM32\DRIVERS\ETC folder where X: is the directory where the Windows operating system is installed.

Example service entry for Broker ETB255:

```
ETB255      3800/tcp
```

## EntireX Broker Services

### CICS (OS/390)

Executing a web enabled Natural application via the Internet running Natural under CICS in an OS/390 operating environment, can use either the WebStar Attach Manager NEEATM or the Broker Services Attach Manager.

If you wish to use the Broker Services Attach Manager, then the following tasks also need to be administered.

#### Broker Attribute File

The following definition is required in the Broker Attribute file and links the Broker Server to the Broker Services SDL.

```
DEFAULTS = SERVICE
CONV-NONACT      = 5M
SERVER-NONACT    = 5M
TRANSLATION      = SAGTCHA
MUOW             = 0
CLASS=AT034C,    SERVER=ATMAN,    SERVICE=ATTACH
```

#### SDL statements

The customer must first set up the Broker Services environment linked to their Broker Server. Once this is place, then the customer should add the appropriate SDL statements for the Broker Services based on the following definitions.

```
DEF BROKER BKR034:SVC249:NET
CLASS AT034C    SERVER ATMAN    SERVICE ATTACH
USERID XXXX    PASSWORD YYYY
WAITTIME 10    RINT 10    RCOUNT 3
```

The DEF BROKER definition above, should already be in the SDL parameter file for the Broker Services as the customer would require this when setting up the link between the Broker Server and the Broker Services. The CLASS AT034C links back to the service defined in the Broker Server Attributes file.

### 3

#### Natural Engineer WebStar

```
DEF PATH R3270
DIR DEVEBVV
ENV CICS METHOD APPC
USERID XXXX PASSWORD YYYY
WAITTIME 10 RINT 10 RCOUNT 4 CONV YES
```

The DEF PATH definition above is defining the CICS region to be used for the submission of the required Natural tasks. DIR DEVEBVV is the LU Name of the CICS region in question.

```
DEF CLASS NEE SERVER NEEEXWB SERVICE INPUT
PATH R3270
REMOTENAME N314
"STACK=(LOGON NEEEXWBX;WEB:WEBP1)"
NATURAL
BROKER BKR034:SVC249:NET
MIN 0 MAX 1000
```

The DEF CLASS definition above is now defining the CICS transaction to use and the appropriate parameters required for starting the Natural transaction. The SERVER NEEEXWB is the Natural application name, which has been web-enabled using WebStar. The PATH R3270 is linking this to the DEF PATH statement selecting the appropriate CICS region. REMOTENAME N314 is specifying the CICS transaction to use. The keyword NATURAL is specifying that this is a Natural session to start. Finally, there is the BROKER BKR034:SVC249:NET linking this to the DEF BROKER statement mentioned earlier.

*Note: If the run time environment utilizes Natural Security, then the Stack information needs to include a user identity and password. For example:*

*STACK=(LOGON NEEEXWBX user-id password;WEB:WEBP1)*

## EntireX Broker Stub

If the web server and EntireX Broker are not on the same machine, then the relevant Broker stub module needs to be installed on the web server.

This will provide the communication between the web server and EntireX Broker.

*Note: For more information refer to section [Broker Stub](#) in the Web Server configuration section.*

## Natural

---

This section describes all the configuration requirements for the Natural Environment.

### Attach Manager NEEATM

#### OS/390 Com-Plete or CICS

As part of the mainframe runtime installation, the INPL will install a library called NEEATM.

This is the WebStar Attach Manager NEEATM for use with OS/390 Com-Plete or CICS. The commands available for the WebStar Attach Manager NEEATM are:

Command	Description
<b>NEEATM START</b>	This will start an asynchronous Natural task under Com-Plete or CICS, which registers with the appropriate broker.
<b>NEEATM END</b>	This will terminate the WebStar Attach Manager NEEATM.
<b>NEEATM STATUS</b>	This will inform the customer of the status of the WebStar Attach Manager NEEATM.

When a client uses the web enabled application, the OS/390 Broker will communicate with the NEEATM task. This will create a separate asynchronous Natural task for each client.

The WebStar Attach Manager NEEATM is controlled by the settings in object NEEATI-S on library NEEATM.

Object	Description
<b>NEEATI-S</b>	This is a subroutine, which contains the necessary default configuration parameters used to communicate between the Natural session and EntireX Broker. The following parameters must be changed to correspond to the individual environment configurations being used:
<b>BROKER-ID</b>	This is the Broker instance that has been defined in EntireX Broker for the application to use.

## 3

## Natural Engineer WebStar

Object	Description
	The default setting is BKR034.
<b>TIME-OUT</b>	Controls length of time for the WebStar Attach Manager before a time out occurs. The default setting is 168H meaning 168 hours, or 1 week.
<b>TRACE-LEVEL</b>	Level of trace diagnostic messages. The default setting is 0. If not 0, then the diagnostic messages are produced.
<b>TRACE-DIR</b>	This is the location of the diagnostic messages. The default setting is *WTL indicating that the messages will be written to the DD CARD APSLOG of the Com-Plete task. If set to *WTO, then the messages will be written to the operator console.
<b>USERID</b>	Natural Security user identity associated with the asynchronous task: NEEATM START.
<b>PASSWORD</b>	Natural Security password associated with the asynchronous task: NEEATM START.

*Note: The parameters 'USERID' and 'PASSWORD' are only required if executing in an environment utilizing Natural Security.*

### Registering Application Servers

With the PC installation, the NEEATM.INI file contains the application servers to be registered with the Broker Server. On the OS/390 and VSE/ESA platforms, the details of which application servers to register are contained in object NEEATA-S on library NEEATM.

NEEATA-S contains a subroutine called `##MAINFRAME`. There is a `DECIDE` statement which contains the parameter settings.

The following Figure 3-1 illustrates the application server parameter settings within object NEEATA-S.

```

DEFINE SUBROUTINE ##MAINFRAME
DECIDE ON FIRST VALUE OF #APP-INDEX
VALUE 1
  #APP-SERVER-CLASS := 'NEE'
  #APP-SERVICE      := 'INPUT'
  #APP-SERVER-NAME  := 'NEEEXWB'
  #APP-PARM         := 'STACK=(LOGON NEEEXWBX;WEB:WEBP1) '
VALUE 2
  #APP-SERVER-CLASS := 'NEE'
  #APP-SERVICE      := 'NEEWEB'
  #APP-SERVER-NAME  := 'NEEWEB'
  #APP-PARM         := 'STACK=(LOGON SHORT;NEEWEB) '
VALUE 3:4
  ASSIGN #APP-RC = 1 /* NOT USED
NONE VALUES
  ASSIGN #APP-RC = 3 /* END OF APPS
END-DECIDE

```

Figure 3-1 Application server parameter settings within object NEEATA-S

The NEEATA-S parameters that need to be set are:

Parameter	Description
<b>#APP-SERVER-CLASS</b>	This is the CLASS name. Possible values: NEE.
<b>#APP-SERVICE</b>	This is the SERVICE name. Possible values: <b>INPUT</b> Used for Natural applications. <b>NEEWEB</b> Used for Short Transaction processing.
<b>#APP-SERVER-NAME</b>	This is the SERVER name. Possible values: <b>'application-name'</b> Used for Natural applications, for example NEEEXWB. <b>NEEWEB</b> Used for Short Transaction processing.

Parameter	Description
<b>#APP-PARM</b>	<p>This is the Natural Stack information used to invoke the application. There are two formats that can be used:</p> <p>1. For WebStar applications:  STACK=(LOGON LLLLLLLL user-id password;WEB:SSSSSSSS)  Where:  <b>LLLLLLLL</b> The name of the Natural library to be used at Natural start up.  <b>user-id</b> A logon user identity.  <b>password</b> A logon password.  <b>WEB:</b> The WebStar run time application control program.  <b>SSSSSSSS</b> The Natural application start up program, for example WEBP1.</p> <p>2. For Short Transactions:  STACK=(LOGON LLLLLLLL user-id password;NEEWEB)  Where:  <b>LLLLLLLL</b> The name of the Natural library to be used at Natural start up.  <b>user-id</b> A logon user identity.  <b>password</b> A logon password.  <b>NEEWEB</b> The WebStar run time Short Transaction control program.</p> <p><i>Note: a logon user identity and password are only required if the run time environment utilizes Natural Security.</i></p>

The user would add a new entry with a different value for each application. The actual value of the #APP-INDEX allocated to an application is not relevant.

The processing repeats calling NEEATA-S with increasing values of #APP-INDEX until it hits the NONE VALUES clause, which gives back a response code of 3.

If the user wishes to temporarily remove an application from being registered they can set the response code (#APP-RC) to 1 rather than remove their definitions from the NEEATA-S module.

For Short Transaction processing, a Natural library containing the Short Transaction objects needs to be defined. This can be a 'dummy' library as the URL used to activate the Short Transaction will contain the Natural library and object names, for example:

```
http://www.myweb.com/scripts/nec.exe/library-name/object-name
```

Figure 3-1 illustrates an example of parameter settings for Short Transaction processing (under Value 2). The Stack information 'STACK=(LOGON SHORT;NEEWEB)' will logon to a library SHORT.

*Note: If the run time environment utilizes Natural Security, then the WebStar run time library SYSNEEW must be defined as a steplib to each Natural library (including dummy ones) from where a Short Transaction object will be executed.*

## NEEINI-S

The WebStar run time object NEEINI-S is the initialization object that provides the connection information between the web enabled Natural application and EntireX Broker.

The object can be found on the supplied WebStar run time library SYSNEEW and should be copied from SYSNEEW to the modification library where the web enabled Natural application resides.

The initialization object needs to be modified to reference the correct EntireX Broker settings being used.

Object	Description
<b>NEEINI-S</b>	This is a subroutine, which contains the necessary default configuration parameters used to communicate between the web enabled Natural application and EntireX Broker. The following parameters must be changed to correspond to the individual environment configurations being used:
<b>BROKER-ID</b>	This is the Broker instance that has been defined in EntireX Broker for the application to use. The default setting is 'ETB255'.
<b>SERVER-NAME</b>	This is the SERVER name specified in the EntireX Broker Attributes File under the server-specific section. The default setting is '????????'.
<b>NAT-ROLLOUT</b>	Controls the amount of resources in use whilst a CICS transaction is waiting for a response from the Broker server. The default setting is 'N'. <i>Note: For more information on this parameter refer to section <a href="#">Reduced Natural resources under CICS (OS/390)</a>.</i>

### **Reduced Natural resources under CICS (OS/390)**

To reduce the amount of resources in use whilst a CICS transaction is waiting for a response from the Broker Server, the parameter **NAT-ROLLOUT** needs to be set to 'Y'.

If set to 'Y', then before the Natural application calls the Broker Server to send data back to the client, it will roll out the Natural thread and call a CICS ASM program called NEEWEB. This then communicates with the Broker Server.

If you would like to activate this process, then a PPT entry is required under the CICS region being used. The load module for the CICS ASM object NEEWEB can be located in the NWSvrs.LOAD data set.

## Natural Parameter Files

### Natural Parameter File for Natural Engineer WebStar

The Natural Parameter file NATENG, used to execute the Natural Engineer WebStar development function under Windows, needs to be modified to include two new work files 13 and 16.

#### Work File 13

This is used to hold the HTML pages for Help maps, which are generated during the modification process using the WebStar development function. The HTML pages will be placed directly on the web server specified.

#### Work File 16

This is used to hold the XSL pages, which are generated during the modification process using the WebStar development function. The XSL pages will be placed directly on the web server specified.

The settings to be used depend on the type of web server being utilized. Below are the recommended default values for each web server supported.

#### Microsoft IIS

Work File 13:

```
Y:\INETPUB\WWWROOT\NEE\~~~~~\#####.HTML
```

Work File 16:

```
Y:\INETPUB\WWWROOT\NEE\~~~~~\#####.XSL
```

#### Apache

Work File 13:

```
Y:\PROGRAM FILES\APACHE GROUP\APACHE\HTDOCS\NEE\~~~~~\#####.HTML
```

Work File 16:

```
Y:\PROGRAM FILES\APACHE GROUP\APACHE\HTDOCS\NEE\~~~~~\#####.XSL
```

**IBM WebSphere**

Work File 13:

Y:\IBM HTTP SERVER\HTDOCS\NEE\~~~~~\#####.HTML

Work File 16:

Y:\IBM HTTP SERVER\HTDOCS\NEE\~~~~~\#####.XSL

**Sun iPlanet**

Work File 13:

Y:\IPLANET\SERVERS\DOCS\NEE\~~~~~\#####.HTML

Work File 16:

Y:\IPLANET\SERVERS\DOCS\NEE\~~~~~\#####.XSL

*Note 1: 'Y:' is the location of the web server.*

*Note 2: If preferred, the destination folders for both the XSL and HTML pages can be routed to another location, for example c:\TEMP, and then copied manually to the web server.*

*Note 3: The '~~~~~' segment of the path name represents the modification library name. The '#####' file name segment represents the object name for the HTML/XSL files generated. Both of these values are dynamically replaced during modification execution.*

### Natural Parameter File for Web Enabled Applications

For each Natural application that has been modified using the WebStar development function, a suitable Natural Parameter file needs to be defined for the application specifying the correct work files, database files and steplib information.

For the WebStar run time function the application Natural Parameter file needs to be modified to include the WebStar run time library SYSNEEW in the steplibs.

Example steplib structure for the sample WebStar application NEEEXWBX:

```
NEEEXWBX  
SYSNEEW
```

The Natural parameter ETID, which is used as an identifier for Adabas-related information needs to be set to ETID=\$.

### Natural Parameter File for OS/390 and VSE/ESA

All the WebStar run time processes referenced by the web enabled application require SYSNEEW library in the steplib chain. This can be achieved either through setting STEPLIB=SYSNEEW or by moving the objects in SYSNEEW to the SYSTEM library.

Other parameters that are required:

```
PC=OFF, MENU=OFF, ETID=$
```

*Note: For more information on these parameters refer to the Natural Operations Manual.*

## Natural Engineer WebStar Initialization Settings

### OS/390 and VSE/ESA

The following Natural Engineer initialization parameters defined in the text member ###CINI need to be specified. They are located in the [MODIFY] group.

*Note: If running using standalone jobs, then the changes need to be made to PDS member CINI found in the NEEvrs.SRCE data set.*

#### 1. **OVERWRITE\_EXISTING=NA**

This parameter must be set 'NA' to stop any existing objects in the Modification library from being overwritten.

#### 2. **WEBSTAR-XSL=WEBSTAR.AAAAAAAA.XSL.PDS**

This parameter needs to be added and will provide the PDS data set name where all generated XSL objects will be output as PDS members.

*Note: 'AAAAAAA' must be specified as shown, this will be dynamically replaced by the Modification process with the application name.*

*For example, if using application NEEEXWB, then the PDS data set will be WEBSTAR.NEEEXWB.XSL.PDS.*

#### 3. **WEBSTAR-HTML=WEBSTAR.AAAAAAAA.HTML.PDS**

This parameter needs to be added and will provide the PDS data set name where all generated HTML objects will be output as PDS members.

*Note: 'AAAAAAA' must be specified as shown, this will be dynamically replaced by the Modification process with the application name.*

*For example, if using application NEEEXWB, then the PDS data set will be WEBSTAR.NEEEXWB.HTML.PDS.*

# 3

## Natural Engineer WebStar

### Natural Security

If the run time environment used to execute NEEATM utilizes Natural Security functionality, then the following must be administered.

1. Define a group named 'WEBSTAR.'
2. Link the group 'WEBSTAR' to Natural libraries NEEATM and SYSNEEW.

*Note: When other applications are added to execute under WebStar, then each application library will need to be linked to the 'WEBSTAR' group.*

3. Define a new user 'NWS442' and assign a default library of NEEATM.
4. Add user 'NWS442' to the 'WEBSTAR' group.
5. Ensure that the user 'NWS442' can sign in to Natural with AUTO=OFF (Natural parameter used for the Natural session being used). Record the password being used.

### User Exit Considerations

The following User Exit objects must be available at run time:

USR0120N  
USR0210N  
USR0320N  
USR0600N  
USR1002N  
USR1005N  
USR1029N  
USR1054N  
USR1055N  
USR1057N  
USR1069N  
USR2013N  
USR2026N  
USR2030N  
USR3025N

*Note: It is recommended that these User Exit objects are copied from SYSEXT to the FNAT library SYSTEM.*

# WEB-ENABLING A NATURAL APPLICATION

## Chapter Overview

---

This chapter explains the processes required to web enable a Natural application using the Natural Engineer add-on component WebStar.

The topics covered are:

- [Applying the WebStar Development Function under Windows](#)
- [Applying the WebStar Development Function under OS/390](#)
- [Preparing the Web Enabled Application for Execution](#)
- [Executing the Web Enabled Natural Application](#)
- [Reporting Mode Considerations](#)
- [Example using Sample Application NEEEXWB](#)

## Applying the WebStar Development Function under Windows

---

This section provides a guide to the various stages required in applying the WebStar development function under Windows.

*Note: This section provides only an overview of the Natural Engineer processes. For more in depth information on each of the processes refer to the individual product manuals. A full list of the available manuals can be found in the section [Related Literature](#).*

### Process Overview

Natural applications are web enabled using the Natural Engineer processes to extract and load the application source code into the Repository. Once loaded, Impact Analysis is executed using a pre-defined set of Impact search criteria. Modifying the application using the Modification process completes the Natural Engineer processes.

This complete cycle of processes will generate the modified objects consisting of Natural subprograms, XSL files to provide the web pages for the presentation layer and HTML files for any Help maps found within the application.

Some manual changes are required to the modified application, to add the required initialization function in order to execute the application via the Internet.

The Natural environment for the application is adapted to utilize the WebStar run time Natural library (SYSNEEW). This is achieved by configuring a Natural Parameter file to define the correct steplib library information and the necessary database file and / or work files required by the application.

With all this in place, the web enabled application can now be cataloged. Once all the correct web server infrastructure is available, the application is ready for execution via the Internet. Additionally, the web enabled application can still be executed from a Natural session in the same way as before the web-enable process.

## Extract the Application

The Extract process will generate a load file containing a neutral view of the application source code, which is used to load the application into the Repository.

The Natural application is extracted using the Natural Engineer Extract process. The application is created using option Application→Open. Application preferences are specified using the option Application→Preferences and then the Extract Selection Criteria are specified using the option Environment→Extract Selection Criteria.

The Natural application can now be extracted by using the option Environment→Extract Source Code. Once this has completed, the Extract process needs to be verified by checking the Extract error log file for any extract errors and producing the Missing Objects report.

Upon satisfactory completion, the load file that has been created can now be loaded into the Repository.

## Load Repository

The Load process will add the neutral view of the application source code into the Repository, this will generate all the information, cross-references and relationships in preparation for the Impact and Modification processes.

The process is executed using option Environment→Load Repository. Once this has completed, the Load process needs to be verified by checking the Load error log file and producing any of the Application Reports, for example the Source Code Summary report.

The application is now ready for the next stage: Impact Analysis.

## Impact Analysis

The Impact Analysis process will identify all the presentation layer statements within an application. The presentation layer statements are any I/O statements, for example: WRITE, DISPLAY or INPUT USING MAP.

Before Impact Analysis can be executed, the required Impact search criteria need to be specified. This is done by first creating an Impact Version using option Analysis→Impact Version, followed by the Impact criteria using option Analysis→Impact Search Criteria.

The specification of the search criteria is simplified with WebStar by a pre-defined criteria set being supplied. This ensures that all the necessary search keywords and values are always used consistently, providing the confidence that all the presentation layer statements are identified.

The supplied search criteria are called WEB.IRE and can be located in the X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\DATA folder. This file can be read in using option File→Open from the Impact Criteria Summary screen.

After the criteria are specified, Impact Analysis can be executed using option Analysis→Impact Execution. Once this has completed, the Impact Analysis can be verified using the option Analysis→Impact Element Maintenance and producing any of the Impact Reports, for example Data Item Impact Inventory report.

### WebStar Modification Preferences

WebStar Modification Preferences can be specified to tailor the presentation of the generated web pages. The preferences must be applied prior to applying the modification.

The WebStar Modification Preferences screen is accessed by using the following menu navigation from the Modification Preferences screen: Options → WebStar Modification Preferences.

The following Figure 4-1 illustrates the WebStar Modification Preferences screen.

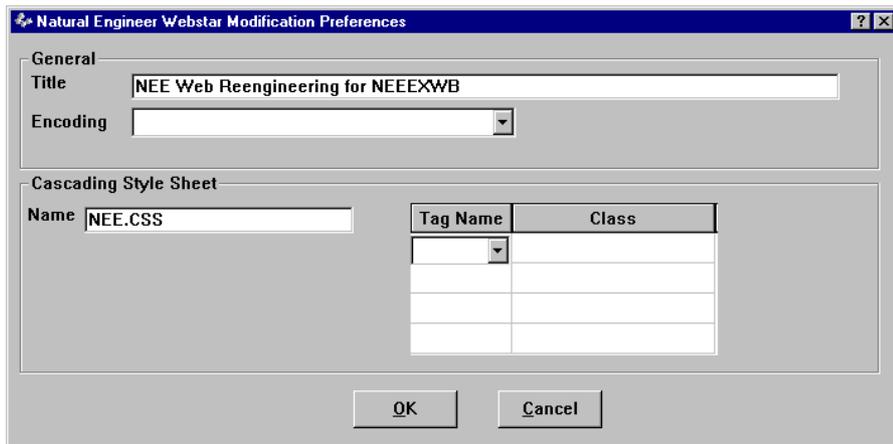


Figure 4-1 WebStar Modification Preferences screen

SCREEN ITEMS	DESCRIPTION
<b>General</b>	General options for the generated web pages.
<b>Title</b>	The name used within each generated web page within the <HEAD></HEAD> tag block.  The default value is: "NEE Web Reengineering for AAAAAAAAA" where AAAAAAAAA is the name of the current opened application.

SCREEN ITEMS	DESCRIPTION
--------------	-------------

<b>Encoding</b>	<p>The name of the character set used to translate any special characters. A value may be selected from the drop list, or a value may be typed in. Possible selections are:</p> <p><b>ISO-10646-UCS-2</b>  <b>ISO-10646-UCS-4</b>  <b>ISO-8859-1</b>  <b>ISO-8859-1-Windows-3.1-Latin-1</b>  <b>US-ASCII</b>  <b>UTF-16</b>  <b>UTF-7</b>  <b>UTF-8</b>  <b>Windows-125n</b></p> <p>If no value is selected or typed in, then the default will be ISO-8859-1.</p> <p><i>Note: The drop list shows only the standard encoding character sets. Any valid encoding character set not in this list may be typed in.</i></p>
<b>Cascading Style Sheet</b>	<p>Cascading Style Sheet options.</p>
<b>Name</b>	<p>The name of the style sheet to be referenced.</p> <p>The default value is NEE.CSS.</p> <p><i>Note: The style sheet needs to be in the NEE folder on the web server being used at execution time.</i></p>
<b>Tag Name</b>	<p>Select the tag name for which properties are to be changed via the style sheet.</p> <p>Possible selection is BODY.</p>
<b>Class</b>	<p>The name of the class within the style sheet, which contains the required properties.</p>

BUTTON NAME	DESCRIPTION
-------------	-------------

<b>OK</b>	Saves the specified WebStar Modification Preferences and return back to the Modification Preferences screen.
<b>Cancel</b>	Exits the WebStar Preferences screen and return back to the Modification Preferences screen.

## Modification

The Modification process will use the Impact Analysis information to modify the existing objects to correctly reference the web presentation layer objects and generate the required web pages (HTML/XSL files), along with the Natural subprograms that will invoke them.

*Note: Both the modification library and the XSL/HTML folder must be empty before the Modification process is invoked. This applies to the first execution and any subsequent re-executions.*

The Modification process is executed using option Modification → Execute Modification for All Objects. Once this has completed, the Modification can be verified by logging onto the modification library and checking the objects.

*Note: If during the Modification execution, the overwrite warning message is presented, always respond using the option "No to All".*

This completes the Natural Engineer WebStar development function.

## Applying the WebStar Development Function under OS/390

---

This section provides a guide to the various stages required in applying the WebStar development function under OS/390.

*Note: This section provides only an overview of the Natural Engineer processes. For more in depth information on each of the processes refer to the individual product manuals. A full list of the available manuals can be found in the section [Related Literature](#).*

### Process Overview

Natural applications are web enabled using the Natural Engineer processes to extract and load the application source code into the Repository. Once loaded, Impact Analysis is executed using a pre-defined set of Impact search criteria. Modifying the application using the Modification process completes the Natural Engineer processes.

This complete cycle of processes will generate the modified objects consisting of Natural subprograms, XSL files to provide the web pages for the presentation layer and HTML files for any Help maps found within the application.

Some manual changes are required to the modified application, to add the required initialization function in order to execute the application via the Internet.

The Natural environment for the application is adapted to utilize the WebStar run time Natural library (SYSNEEW) by copying it to the FUSER library SYSTEM. This is achieved by configuring a Natural Parameter file to define the necessary database file and / or work files required by the application.

With all this in place, the web enabled application can now be cataloged. Once all the correct web server infrastructure is available, the application is ready for execution via the Internet. Additionally, the web enabled application can still be executed from a Natural session in the same way as before the web-enable process.

## Extract the Application

The Extract process will generate a load file containing a neutral view of the application source code, which is used to load the application into the Repository.

The Natural application is extracted using the Natural Engineer Extract process. The application is created using option 'S' (Select an Application) from the Application Menu screen. Application preferences are specified using option 'P' (Preferences) from the Application Menu screen and then the Extract Selection Criteria are specified using option 'S' (Extract Selection Criteria) from the Environment Menu screen.

The Natural application can now be extracted by using option 'E' (Extract Source Code) from the Environment Menu screen. Once this has completed, the Extract process needs to be verified by checking the Extract error log file for any extract errors and producing the Missing Objects report.

Upon satisfactory completion, the load file that has been created can now be loaded into the Repository.

## Load Repository

The Load process will add the neutral view of the application source code into the Repository, this will generate all the information, cross-references and relationships in preparation for the Impact and Modification processes.

The process is executed using option 'L' (Load Repository) from the Environment Menu screen. Once this has completed, the Load process needs to be verified by checking the Load error log file and producing any of the Application Reports, for example the Source Code Summary report.

The application is now ready for the next stage: Impact Analysis.

## Impact Analysis

The Impact Analysis process will identify all the presentation layer statements within an application. The presentation layer statements are any I/O statements, for example: WRITE, DISPLAY or INPUT USING MAP.

Before Impact Analysis can be executed, the required Impact search criteria need to be specified. This is done by first creating an Impact Version using option 'V' (Impact Versions) from the Impact Analysis Menu screen, followed by the Impact criteria using option 'C' (Impact Search Criteria) from the Impact Analysis Menu screen.

The specification of the search criteria is simplified with WebStar by a pre-defined criteria set being supplied. This ensures that all the necessary search keywords and values are always used consistently, providing the confidence that all the presentation layer statements are identified.

The supplied search criteria are called WEBSTAR and can be read in using option 'PF4' (GetSa) from the Impact Criteria Summary screen.

After the criteria are specified, Impact Analysis can be executed using option 'E' (Impact Execution) from the Impact Analysis Menu screen. Once this has completed, the Impact Analysis can be verified using option 'M' (Impact Element Maintenance) from the Impact Analysis Menu screen and producing any of the Impact Reports, for example Data Item Impact Inventory report.

## Environment Administration

Before the Modification is applied, there are several administration tasks that need to be performed for the environment being used.

### ###CINI Parameters

The following Natural Engineer initialization parameters defined in the text member ###CINI need to be specified. They are located in the [MODIFY] group.

*Note: If running using standalone jobs, then the changes need to be made to PDS member CINI found in the NEEvrs.SRCE data set.*

#### 1. **OVERWRITE\_EXISTING=NA**

This parameter must be set 'NA' to stop any existing objects in the Modification library from being overwritten.

#### 2. **WEBSTAR-XSL=WEBSTAR.AAAAAAAAA.XSL.PDS**

This parameter needs to be added and will provide the PDS data set name where all generated XSL objects will be output as PDS members.

*Note: 'AAAAAAAA' must be specified as shown, this will be dynamically replaced by the Modification process with the application name.*

*For example, if using application NEEEXWB, then the PDS data set will be WEBSTAR.NEEEXWB.XSL.PDS.*

#### 3. **WEBSTAR-HTML=WEBSTAR.AAAAAAAAA.HTML.PDS**

This parameter needs to be added and will provide the PDS data set name where all generated HTML objects will be output as PDS members.

*Note: 'AAAAAAAA' must be specified as shown, this will be dynamically replaced by the Modification process with the application name.*

*For example, if using application NEEEXWB, then the PDS data set will be WEBSTAR.NEEEXWB.HTML.PDS.*

## 4

### Natural Engineer WebStar

#### XSL and HTML PDS Data Sets

Two PDS data sets need to be created for the generated XSL and HTML objects:

```
WEBSTAR.AAAAAAAA.XSL.PDS
```

```
WEBSTAR.AAAAAAAA.HTML.PDS
```

*Note: 'AAAAAAA' must be specified as the application name being web-enabled.*

*For example, if using application NEEEXWB, then the PDS data sets will be WEBSTAR.NEEEXWB.XSL.PDS and WEBSTAR.NEEEXWB.HTML.PDS.*

Each PDS data set should have an Organization of '**PO**', a Record Format of '**VB**' and a Record Length of '**260**' specified. All other characteristics can be specified as per site standards.

The application is now ready for the next stage: Modification.

#### Modification

The Modification process will use the Impact Analysis information to modify the existing objects to correctly reference the web presentation layer objects and generate the required web pages (HTML/XSL files), along with the Natural subprograms that will invoke them.

*Note: Both the modification library and the XSL/HTML PDS data sets must be empty before the Modification process is invoked. This applies to the first execution and any subsequent re-executions.*

The Modification process is executed using option '**A**' (Execute Modification for All Objects) from the Modification Menu screen. Once this has completed, the Modification can be verified by logging onto the modification library and checking the objects.

*Note: If during the Modification execution, an error is produced for work file 13 or 16, then it is because the XSL and/or HTML PDS data sets do not exist.*

This completes the Natural Engineer WebStar development function.

## Preparing the Web Enabled Application for Execution

---

Before the web enabled application can be executed there are a few administrative tasks required.

### 1. Ensure that the modification library is complete.

The Modification process will only create objects from an application, where the Impact Analysis process has identified presentation layer statements. Any application objects that contain no presentation layer statements will not be copied into the modification library. These will need to be copied across from the base application library, or migrate the contents of the modification library to an appropriate library to meet your change control requirements.

### 2. Check the Application Exit points.

The web enabled application exit points must issue a TERMINATE statement in order to ensure a clean and complete exit from the Natural Server session.

If the exit points within the application do not issue the TERMINATE statement, then when the web enabled application is stopped from the browser, the application 'drops' into Natural leaving the Natural Server session hanging in EntireX Broker.

The WebStar run time process interprets the TERMINATE statement as FETCH 'NEETERM'. This will execute the correct controlled close down procedure.

An alternative to issuing TERMINATE statements in all exit points is to code the following for each exit point:

```
IF *CONVID NE 0
    FETCH 'NEETERM'
END-IF
```

# 4

## Natural Engineer WebStar

### 3. Add the necessary WebStar initialization object.

The web enabled application needs to have added an initialization object, which is required to start the application when executed via the Internet.

The initialization object can be found on the supplied WebStar run time library SYSNEEW and should be copied from SYSNEEW to the modification library.

Object	Description
<b>NEEINI-S</b>	This is a subroutine, which contains the necessary default configuration parameters used to communicate between the Natural session and EntireX Broker. The following parameters must be changed to correspond to the individual environment configurations being used:  <b>BROKER-ID</b> This is the Broker instance that has been defined in EntireX Broker for the application to use. The default setting is ETB255.  <b>SERVER-NAME</b> This is the name of the server that has been defined in the Broker Attribute File.  <b>TRACE-LEVEL</b> Level of trace diagnostic messages. If set to 0 (default), no messages are produced. Any other value will produce the messages.  <b>TRACE-DIR</b> This is the location of the diagnostic log files. The default setting is c:\temp.

*Note: For more information on the EntireX Broker settings refer to section [EntireX Broker](#) in Chapter 3.*

*Note: The web enabled application can still be executed using a Natural session by invoking the application start up program.*

### 4. WebStar run time library SYSNEEW.

#### Under Windows

The Natural Parameter file being used for the web enabled application needs to have the WebStar run time library SYSNEEW added as a steplib library for the application.

*Note: For more information refer to section [Natural Parameter File for Web Enabled Applications](#) in Chapter 3.*

The web enabled application can now be cataloged.

#### **Under OS/390**

Copy the WebStar run time library SYSNEEW to FUSER library SYSTEM. The web enabled application can now be cataloged.

### **5. Add any required Web files to the Web Server.**

#### **Under Windows**

The web enabled application folder on the web server (containing the generated XSL and HTML files) needs to have the 'templ.html' file copied from the NEE folder on the web server. This file is used to populate the left frame at run time.

*Note: Any other customized web files that are required by the web enabled application at run time should also be added to this folder.*

#### **Under OS/390**

The PDS data sets containing the generated XSL and HTML objects need to be copied to the web server. File extensions '.xsl' and '.html' need to be added to each of the PDS member names during the copy. Also, any case dependencies for the destination web server need to be observed.

*Note: If using IBM WebSphere under OS/390 Unix System Services, an example copy job called 'M2U' is supplied, which can be modified to site standards.*

*It will copy the generated XSL and HTML PDS members, adding the relevant file extensions and applying the correct case dependency.*

*It can be found in the NWSvrs.WEBS load file. The copy job must be copied to the OS/390 Unix System Services environment and executed from there.*

The web enabled application folder on the web server (containing the generated XSL and HTML files) needs to have the 'templ.html' file copied from the NEE folder on the web server. This file is used to populate the left frame at run time.

## 4

### Natural Engineer WebStar

#### 6. Natural Security checking for WebStar logon page.

If the run time environment used to execute the web enabled application utilizes Natural Security functionality, then it is possible to process and validate the user identity and password entered in the WebStar logon page against Natural Security definitions.

The following needs to be administered:

1. Copy object NSC---P from the FNAT library SYSSEC to the FNAT or FUSER library SYSTEM.
2. The Natural Security profile for the web enabled application must have SYSLIBS added as a steplib library.
3. The WebStar initialization object NEEINI-S needs to have the following clause added to the :

```
VALUE 'SECURITY'  
ASSIGN #P-VALUE = 'Y'
```

*Note: The current version of WebStar only supports passwords entered in upper case.*

## Executing the Web Enabled Natural Application

---

This section describes how to execute the web enabled Natural applications via the Internet.

*Note: If the web enabled application is to be executed via the SMARTS HTTP server using the Natural CGI program then EntireX Broker and Attach Manager (NEEATM) are not required.*

### EntireX Broker

The EntireX Broker Instance being used for the application must be invoked using the EntireX Broker Administration function and selecting the Broker to be used, for example ETB255.

### Attach Manager (NEEATM)

#### Windows

The Attach Manager must be invoked by executing object NEEATM.EXE. This is located in the X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\WEB\NEEATM folder (where X: is the directory where Natural Engineer has been installed). Typically this would be invoked using the START → RUN command line.

To start the Attach Manager:

```
"X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\WEB\NEEATM\NEEATM.EXE"
```

To terminate the Attach Manager:

```
"X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\WEB\NEEATM\NEEATM.EXE  
END"
```

## 4

### Natural Engineer WebStar

#### CICS and Com-Plete

Log on to Natural and then log on to library NEEATM. The following commands can be used:

To start the Attach Manager:

```
NEEATM START
```

To terminate the Attach Manager:

```
NEEATM END
```

To determine the status of the Attach Manager:

```
NEEATM STATUS
```

#### Sun Solaris 8

*Note: The documentation relates to the Unix environment where case sensitivity is important. Any described input settings should be treated as read.*

The Attach Manager must be invoked by executing object NEEATM.EXE. This is located in the '/sag/nws/neeatm' directory. Typically this would be invoked using the Terminal Window command line.

To start the Attach Manager:

```
./neeatm.exe > neeatm.log 2>&1 &
```

To terminate the Attach Manager:

```
./neeatm.exe END
```

## Web Browser

### General

Open a web browser and type in the URL based on the following:

```
http://www.myweb.com/nee/webstar.html
```

where 'www.myweb.com' is the address of the web server, or a TCP/IP address.

This will now display the WebStar logon page. The User-id and password are input and an application selected. The next web page will be the first screen of the selected application.

### SMARTS HTTP Server

If using the SMARTS HTTP Server environment, the web enabled application can be invoked using two different methods:

#### 1. Using the WebStar web interface program NEEC.

The following steps are required to execute the web enabled application:

**Step 1** The FORM element within the WebStar TEMPM.HTML template page and the generated XSL template pages used by the web enabled application need to be modified to reflect the correct script reference.

*Note: The generated XSL template pages are located in the web enabled application PDS data set.*

This can be applied in two different ways:

1. Manually.

Modify the FORM element within the template pages to point to the WebStar web interface program NEEC:

```
<form name="logon" method="post" action="/cgi/need">
```

## 4

### Natural Engineer WebStar

2. Using the supplied SMARTS user exit HAANUXIT with environment variables.

Add the following environment variable pairs to the SMARTS HTTP PDS member containing the environment variables being utilized.

For the local site standard HLQ:

```
NEEALIAS=/nee
```

```
NEEALIAS2=/sitehlq/nee
```

*Note: These are only required if the local site standards do not permit the use of NEE.*

The WebStar web interface program NEEC:

```
NEESCRIPALIAS=/scripts/nec.exe
```

```
NEESCRIPTSALIAS2=/cgi/nec
```

**Step 2** Open a web browser and type in the URL based on the following:

```
http://www.myweb.com/nee/webstar.html
```

where 'www.myweb.com' is the address of the web server, or a TCP/IP address.

This will now display the WebStar logon page. The User-id and password are input and an application selected. The next web page will be the first screen of the selected application.

## 2. Using the Natural web interface program.

This will invoke the Natural CGI program to provide the interface between Natural and the web server. The following steps are required to execute the web enabled application:

**Step 1** The FORM element within the WebStar TEMPM.HTML template page and the generated XSL template pages used by the web enabled application need to be modified to point to the Natural CGI program.

*Note: The generated XSL template pages are located in the web enabled application PDS data set.*

This can be applied in two different ways:

1. Manually.

Modify the FORM element within the template pages to point to the Natural CGI program and map the WebStar control program NEEAPL:

```
<form name="logon" method="post" action="/natcgi/sysneew/neeapl">
```

The WebStar control program NEEAPL provides the link between the WebStar logon page and the Natural CGI, passing the User id, password and application selection information.

2. Using the supplied SMARTS user exit HAANUXIT with environment variables.

Add the following environment variable pairs to the SMARTS HTTP PDS member containing the environment variables being utilized.

For the local site standard HLQ:

```
NEEALIAS=/nee
```

```
NEEALIAS2=/sitehlq/nee
```

*Note: These are only required if the local site standards do not permit the use of NEE.*

The WebStar web interface program NEEC:

```
NEESCRIPALIAS=/scripts/neec.exe
```

```
NEESCRIPTSALIAS2=/natcgi/sysneew/neeapl
```

## 4

### Natural Engineer WebStar

**Step 2** Check that the correct WebStar run time NEE PDS data set path is specified.

This will locate the WebStar run time function components to the global initialization settings found in the WebStar run time subroutine NEEAPI-S on the WebStar run time library SYSNEEW.

The NEE PDS data set path is specified to a variable #P-VALUE when processing the input parameter value 'XSL-URL'.

The default value is set as:

```
VALUE 'XSL-URL'  
    ASSIGN #P-VALUE = '/nee'
```

If your site does not permit NEE to be used as the HLQ, then this value needs to be modified:

```
VALUE 'XSL-URL'  
    ASSIGN #P-VALUE = '/sitehlq/nee'
```

**Step 3** Add the web enabled application information to list of applications.

The web enabled application information needs to be added to the applications list found in the WebStar run time subroutine NEEAPL-S on the WebStar run time library SYSNEEW. This list maintains the list of applications that will appear on the WebStar logon screen.

Each entry in the list must be defined as follows:

Parameter	Description
VALUE	This is the application sequence number in the list. Valid range is 1 to 1000.
#APP-TITLE	This is the name of the application that is displayed on the WebStar logon page.
#APP-NATLIB	The Natural library name for the application, for example NEEEXWBX.
#APP-NATOBJ	The Natural applications start up program, for example WEBP1.
#APP-NEE-DIR	The name of the PDS data set containing the generated XSL template pages, for example NEEEXWBX.
#APP-RC	Set a return code of 1 to remove the application from the list. This parameter is optional.

**Step 4** Open a web browser and type in the URL based on the following:

`http://www.myweb.com/nee/webstar.html`

where 'www.myweb.com' is the address of the web server, or a TCP/IP address.

*Note: If your site does not permit NEE to be used as the HLQ, then the URL needs to be modified to `http://www.myweb.com/sitehlq/nee/webstar.html`.*

This will now display the WebStar logon page. The User-id and password are input and an application selected. The next web page will be the first screen of the selected application.

#### **Restrictions with SMARTS HTTP Server**

Using the SMARTS HTTP server has the following restrictions:

1. Using the Natural Web Interface program will only support XSL template pages. This means that if the web-enabled application uses any map help routines, which will generate HTML template pages during the WebStar modification process, these will not be available.
2. The SMARTS HTTP server has no 'cache' protocol and will therefore, always download any requested URL each time.

## Reporting Mode Considerations

---

Any Reporting Mode objects within an application will need to be modified to include certain Structured Mode characteristics prior to applying the WebStar development function.

### Mode Conversion

The Mode conversion option will modify any Reporting Mode objects to include the required Structured Mode characteristics. It is invoked by using option Utilities→Mode Conversion. This option will apply the following changes to the Reporting Mode objects:

- Add Define Data statement.
- Add Views.
- Add Labels and back references.
- Rearrange STORE/UPDATE statements.
- Split multi statement lines.

### Natural Engineer Process Overview

To web enable a Natural application containing any Reporting mode objects, then the following steps are required using Natural Engineer with the WebStar add-on component:

1. Extract and Load the Natural application into the Repository.
2. If you have AIVs within your system use the option Utilities→Mode Conversion→Options→Create GDA to create a GDA.
3. Apply Mode conversion. If you have created a GDA specify the name of the GDA to be inserted during the conversion process when prompted.
4. Extract and Load the converted application into the Repository.
5. Execute Impact Analysis using the predefined WebStar Impact search criteria.
6. Apply the WebStar modification.

## Example using Sample Application: NEEEXWB

---

The following example illustrates both the WebStar development and run time functions running under the Windows operating system using the sample application NEEEXWB.

The example will show the steps required within Natural Engineer to modify the sample application NEEEXWB. It will also show the run time environment settings and finally, illustrate the web enabled application executing via the Internet.

The sample application NEEEXWB contains 8 objects that provide a simple Employees administration system. The application references a single database file EMPLOYEEES.

**Step 1** Create a new application called **NEEEXWB**. This is achieved by using the option Application→Open from the main Natural Engineer screen.

The following Figure 4-2 illustrates the creation of application NEEEXWB.

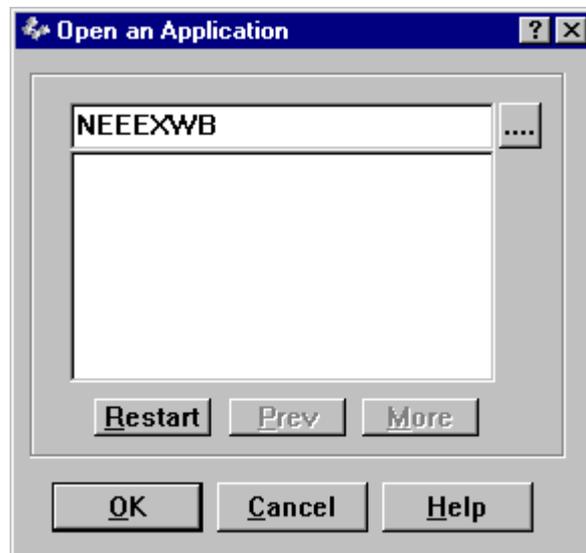


Figure 4-2 Create application NEEEXWB

# 4

## Natural Engineer WebStar

**Step 2** Application Preferences. These do not need to be specified for this example as we are using the application name NEEEXWB, which is the same as the FUSER Natural library.

The Extract process will automatically extract the source code from the Natural library NEEEXWB. The modification library name will be automatically generated to be NEEEXWBX (Natural Engineer will add 'X' to the last character of the application name).

**Step 3** Specify the Extract Selection Criteria. This is achieved by using the option Environment → Extract Selection Criteria from the main Natural Engineer screen.

The wildcard '\*' is specified as the Start Object in Row 1, this will extract all objects from the Natural library NEEEXWB.

The following Figure 4-3 illustrates the specification of the Extract Selection Criteria.

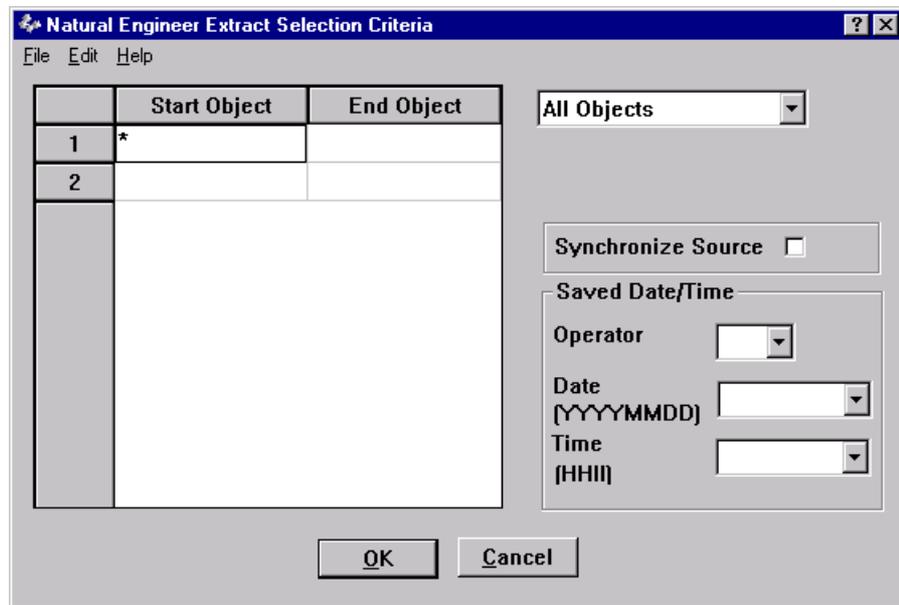


Figure 4-3 Specify Extract Selection Criteria

**Step 4** Extract the application source code. This is achieved by using the option Environment → Extract Source Code.

The Extract batch process will now execute, extracting application source code into the work file **NEEEXWB.OUT**. All the data in this file will contain a 'neutral view' of the application.

**Step 5** Load the Repository. This can be achieved by using the option Environment → Load Repository.

The Load batch process will now execute, loading the 'neutral view' data in the **NEEEXWB.OUT** file into the Repository.

**Step 6** Review the Load process. To review that all the correct objects for the application have been extracted and loaded into the Repository, the Object Summary Report will be produced. This is achieved by using option Application Reports → Object Summary.

The following Figure 4-4 illustrates the Object Summary report for application NEEEXWB.

<i>Object Summary</i>						
Application: NEEEXWB						
Object Type	Object Name	Total Objects	Total Lines	Steplib Application	Save Time	Load Time
Data Defn. Module	EMPLOYEES		37			2003/02/14 16:00:52
	<b>Totals:</b>	<b>1</b>	<b>37</b>			
Global Data Area	WEBG1		3		2001/12/19 19:43:00	2003/02/14 16:00:45
	<b>Totals:</b>	<b>1</b>	<b>3</b>			
Map	WEBM1		145		2002/01/08 09:19:00	2003/02/14 16:00:46
	WEBM2		150		2002/01/08 09:19:00	2003/02/14 16:00:47
	WEBM3		128		2002/01/08 09:19:00	2003/02/14 16:00:48
	WEBM4		129		2002/01/08 09:19:00	2003/02/14 16:00:49
	WEBM5		151		2002/01/08 09:19:00	2003/02/14 16:00:49
	<b>Totals:</b>	<b>5</b>	<b>703</b>			
Subprogram	WEBN1		80		2001/12/21 09:31:00	2003/02/14 16:00:50
	<b>Totals:</b>	<b>1</b>	<b>80</b>			
Program	WEBP1		80		2002/03/13 17:37:00	2003/02/14 16:00:51
	<b>Totals:</b>	<b>1</b>	<b>80</b>			
<b>Grand Total:</b>		<b>9</b>	<b>903</b>			

Figure 4-4 Object Summary Report for application NEEEXWB

## 4

### Natural Engineer WebStar

**Step 7** Create an Impact Version. This can be achieved by using the option Analysis→Impact Version from the main Natural Engineer screen.

Impact version 1 will be created with a comment of "**Web enable application NEEEXWB**".

The following Figure 4-5 illustrates the Impact Version screen with version 1 details.

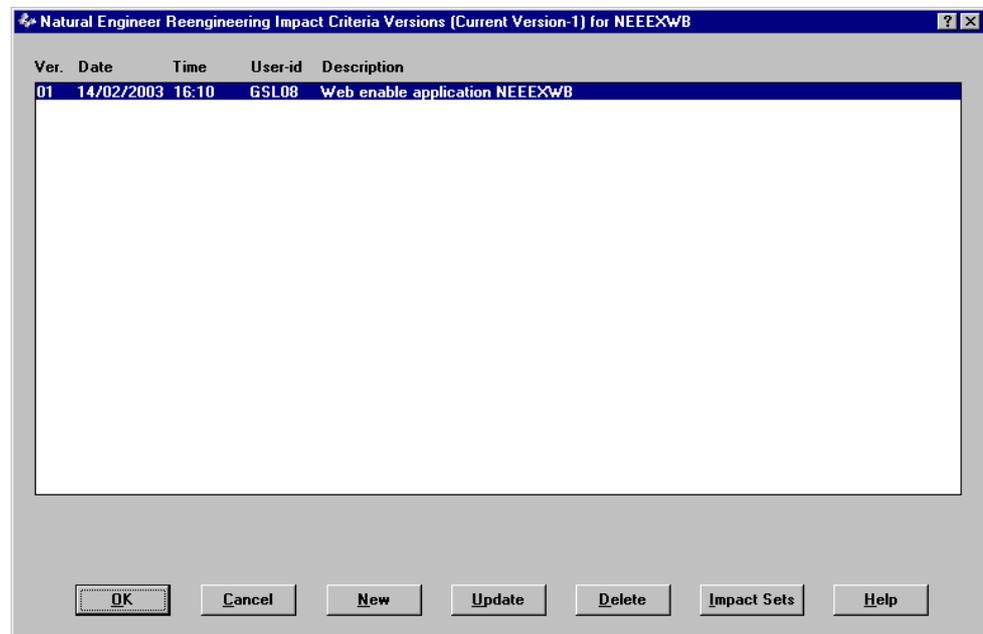


Figure 4-5 Impact Version 1 created

**Step 8** Create the Impact Search Criteria. This can be achieved by using the option Analysis→Impact Search Criteria from the main Natural Engineer screen.

The Impact Search Criteria to be used for WebStar comes supplied as a pre-defined set in file **WEB.IRE**, which is located in X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\DATA folder (where X: is the directory on which Natural Engineer has been installed).

The criteria are specified using the option Analysis→Impact Search Criteria from the main Natural Engineer screen. This will open the Impact Criteria Summary screen.

The **WEB.IRE** data can be read in by using option File→Open from the Impact Criteria Summary screen, locating the file and using the Open button.

The following Figure 4-6 illustrates the Open File screen with the WEB.IRE file selected.

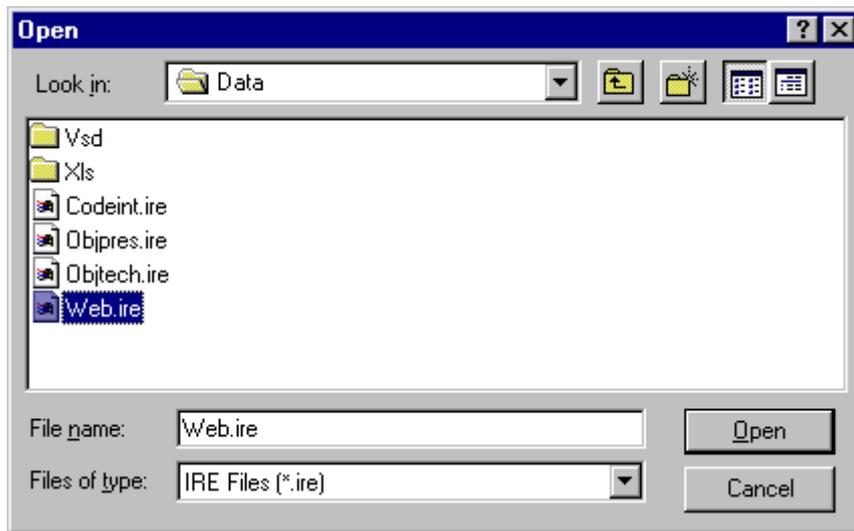


Figure 4-6 Selecting the file WEB.IRE

The following Figure 4-7 illustrates the Impact Criteria Summary screen with all the WebStar Impact Search Criteria.

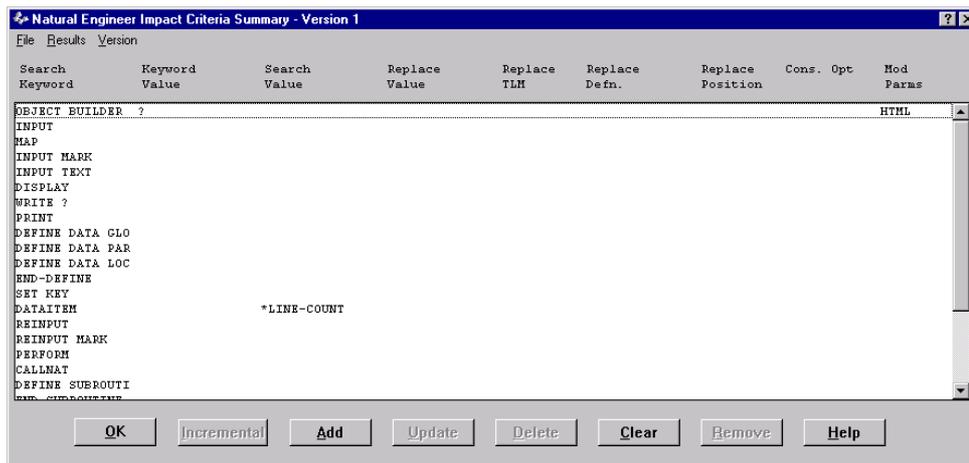


Figure 4-7 WebStar Impact Search Criteria

# 4

## Natural Engineer WebStar

**Step 9** Execute Impact Analysis. This is achieved by using option Analysis → Impact Execution from the main Natural Engineer screen.

The Impact batch process will now execute. The Impact process will be applying each of the criteria to each of the application objects searching for all presentation layer statements.

**Step 10** Review the Impact Analysis. This is achieved by using option Analysis → Impact Element Maintenance from the main Natural Engineer screen.

There will be 2 objects impacted for presentation layer statements in the NEEEXWB application: **WEBP1** and **WEBN1**.

The following Figure 4-8 illustrates the Impact Element Maintenance screen displaying the Impact results for object WEBP1.

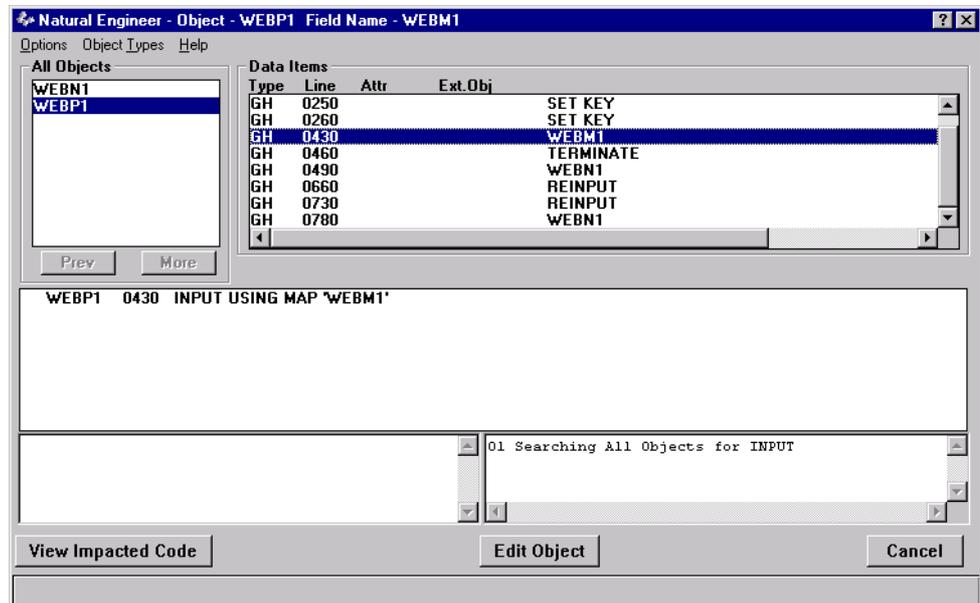


Figure 4-8 Impact Element Maintenance screen displaying the Impact results for WEBP1

**Step 11** Review the Modification criteria. This is achieved by using option Modification → Modification Element Maintenance from the main Natural Engineer screen.

The following Figure 4-9 illustrates the Modification Element Maintenance screen displaying the Modification criteria for object WEBP1.

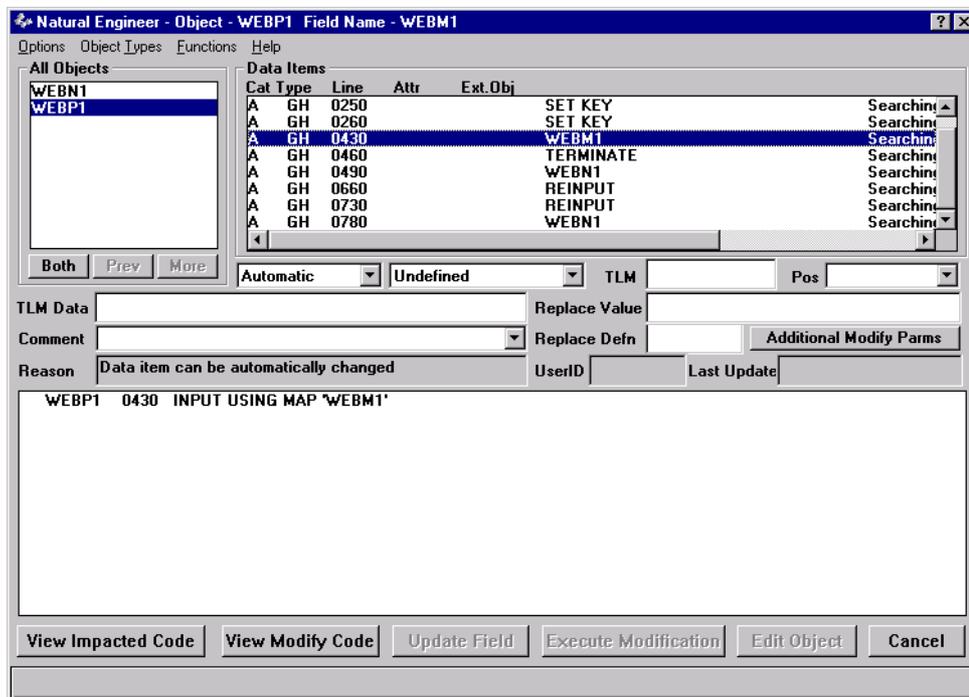


Figure 4-9 Modification Element Maintenance screen displaying the Modification criteria for WEBP1

## 4

### Natural Engineer WebStar

**Step 12** Apply the Modification. This is achieved by using option Modification → Execute Modification for All Objects from the main Natural Engineer screen.

*Note: Prior to executing the Modification, both the modification library and the folders, which will contain the XSL and HTML files must be checked to ensure they are empty.*

The Modification batch process will now execute. The Modification process will apply the Modification criteria to each of the impacted objects, generating subprograms and XSL/HTML files for each presentation layer statement.

*Note: If during the Modification execution, the overwrite warning message is presented, the option "No to All" should always be selected.*

**Step 13** Review the modified and generated Natural objects. This is achieved by opening a Natural session and logging onto the modification library NEEEXWBX.

There are 7 objects in the library, the modified program WEBP1 and subprogram WEBN1, plus 5 new subprograms for each of the presentation layer statements found in the application.

The following Figure 4-10 illustrates the object list for the modification library NEEEXWBX.

Name	Type	User ID	Source Date	Source Size	Catalog Date	Catalog Size	Mode
WEBN1	Subprogram	GSL08	2003-02-14 16:16	8042			Structured
WEBN1N01	Subprogram	GSL08	2003-02-14 16:15	8781			Structured
WEBN1N02	Subprogram	GSL08	2003-02-14 16:15	8781			Structured
WEBN1N03	Subprogram	GSL08	2003-02-14 16:15	8781			Structured
WEBN1N04	Subprogram	GSL08	2003-02-14 16:16	8781			Structured
WEBP1	Program	GSL08	2003-02-14 16:17	3761			Structured
WEBP1N01	Subprogram	GSL08	2003-02-14 16:16	4629			Structured

Figure 4-10 Object list for modification library NEEEXWBX

**Step 14** Review the modified object WEBP1. This will show the WebStar modifications that have been applied.

The following Figure 4-11 illustrates the source code listing for modified object WEBP1 with the modified code shown in bold.

```

0010 DEFINE DATA GLOBAL USING WEBG1
0020 LOCAL
0030 01 EMP VIEW OF EMPLOYEES
0040 02 PERSONNEL-ID
0050 02 NAME
0060 02 FIRST-NAME
0070 * 02 MIDDLE-I
0080 02 MAR-STAT
0090 02 SEX
0100 01 #MAP-TABLE (15)
0110 02 #MPERSONNEL-ID (A8)
0120 02 #MNAME (A20)
0130 02 #MFIRST-NAME (A20)
0140 02 #MMIDDLE-I (A1)
0150 02 #MMAR-STAT(A1)
0160 02 #MSEX (A1)
0170 01 #ACTION (A1/15)
0180 01 #I (N3)
0190 01 #J (N3)
0200 01 #START (A8)
0210 /*
0220 LOCAL USING NEESTD-A /* NEE MODIFIED
0230 LOCAL USING NEECON-L /* NEE MODIFIED
0240 LOCAL USING NEEOBJ-L /* NEE MODIFIED
0250 LOCAL /* NEE MODIFIED
0260 01 #@NEE-SYSTEM-VARIABLES /* NEE MODIFIED
0270 02 #@NEE-SYS-LIBRARY-ID (A8) /* NEE MODIFIED
0280 LOCAL USING NEESYS-L /* NEE MODIFIED
0290 END-DEFINE
0300 /* SET KEY PF1 = 'HELP'
0310 /* SET KEY PF2 NAMED 'RETN'
0320 CALLNAT 'NEEETU-N' 'U ' #@NEE-SYS-USER /* NEE MODIFIED
0330 SET KEY PF3 NAMED 'END'
0340 MOVE #@NEE-PFKEY-ACTIVE TO #@NEE-KEY-PGMSENSE-STATUS(3) /* NEE MODIFIED
0350 MOVE #@NEE-KEY-PGMSET TO #@NEE-KEY-PGMSENSE-SET(3) /* NEE MODIFIED
0360 MOVE 'END' TO #@NEE-KEY-NAME(3) /* NEE MODIFIED
0370 SET KEY PF4 NAMED 'ADD'
0380 MOVE #@NEE-PFKEY-ACTIVE TO #@NEE-KEY-PGMSENSE-STATUS(4) /* NEE MODIFIED
0390 MOVE #@NEE-KEY-PGMSET TO #@NEE-KEY-PGMSENSE-SET(4) /* NEE MODIFIED
0400 MOVE 'ADD' TO #@NEE-KEY-NAME(4) /* NEE MODIFIED
0410 REPEAT
0420 RESET #I
0430 RESET #MPERSONNEL-ID(*) #MNAME(*) #MFIRST-NAME(*)
0440 #MMAR-STAT(*) #MSEX(*)
0450 READ (14) EMP BY PERSONNEL-ID FROM #START

```

```

0460  RESET #START
0470  ADD 1 TO #I
0480  #MPERSONNEL-ID(#I):=PERSONNEL-ID
0490  #MNAME(#I):=NAME
0500  #MFIRST-NAME(#I):=FIRST-NAME
0510  * #MMIDDLE-I(#I):=MIDDLE-I
0520  #MMAR-STAT(#I):=MAR-STAT
0530  #MSEX(#I):=SEX
0540  END-READ
0550  RESET #I
0560  REPEAT
0570  INCLUDE NEEQ /* NEE MODIFIED
0580  INPUT USING MAP 'WEBM1'
0590  IF *CONVID GT 0 /* NEE MODIFIED
0600  CALLNAT 'WEBP1N01' #@NEE-STD-PARMS /* NEE MODIFIED
0610  #ACTION(1:15) #MAP-TABLE.#MFIRST-NAME(1:15) #MAP-TABLE.#MMAR-STAT(1:15)
/* NEE MODIFIED
0620  #MAP-TABLE.#MMIDDLE-I(1:15) #MAP-TABLE.#MNAME(1:15) /* NEE MODIFIED
0630  #MAP-TABLE.#MPERSONNEL-ID(1:15) #MAP-TABLE.#MSEX(1:15) #START /* NEE
MODIFIED
0640  #@NEE-SYS-LIBRARY-ID #@NEE-SYS-DAT4U #@NEE-SYS-USER #@NEE-SYS-PROGRAM
/* NEE MODIFIED
0650  END-IF /* NEE MODIFIED
0660  IF *PF-KEY = 'PF3'
0670  /* ESCAPE BOTTOM IMMEDIATE
0680  /* TERMINATE /* NEE OLD CODE
0690  RELEASE STACK /* NEE MODIFIED
0700  FETCH 'NEETERM' /* NEE MODIFIED
0710  END-IF
0720  IF *PF-KEY = 'PF4'
0730  CALLNAT 'WEBN1' ' ' ' 'A'
0740  #@NEE-STD-PARMS /* NEE MODIFIED
0750  ESCAPE BOTTOM IMMEDIATE
0760  END-IF
0770  IF #START NE ' '
0780  /*
0790  ESCAPE BOTTOM
0800  END-IF
0810  FOR #I = 1 TO 14
0820  IF #MPERSONNEL-ID(1) NE ' '
0830  IF #ACTION(#I) NE ' '
0840  #J:=1
0850  END-IF
0860  END-IF
0870  END-FOR
0880  IF #J = 1
0890  IGNORE
0900  ELSE
0910  MOVE

```

```

0920 'PLEASE ENTER A ACTION CODE'
0930 TO #@NEE-REINPUT-LINE /* NEE MODIFIED
0940 RESET #@NEE-REINPUT-PARM(*) /* NEE MODIFIED
0950 INCLUDE NEEQ /* NEE MODIFIED
0960     REINPUT 'PLEASE ENTER A ACTION CODE'
0970     END-IF
0980     #J:=0
0990     FOR #I = 1 TO 14
1000         IF #ACTION(#I) = ' ' OR = 'M' OR = 'D' OR = 'P'
1010             IGNORE
1020         ELSE
1030     MOVE
1040 'PLEASE ENTER A ACTION CODE OF D OR M OR P'
1050 TO #@NEE-REINPUT-LINE /* NEE MODIFIED
1060 RESET #@NEE-REINPUT-PARM(*) /* NEE MODIFIED
1070 INCLUDE NEEQ /* NEE MODIFIED
1080     REINPUT 'PLEASE ENTER A ACTION CODE OF D OR M OR P'
1090     END-IF
1100     END-FOR
1110     FOR #I = 1 TO 14
1120         IF #ACTION(#I) NE ' '
1130             CALLNAT 'WEBN1' #MPERSONNEL-ID(#I) #ACTION(#I)
1140 #@NEE-STD-PARMS /* NEE MODIFIED
1150     END-IF
1160     END-FOR
1170 RESET #ACTION(*)
1180 END-REPEAT
1190 IF *PF-KEY = 'PF3'
1200     ESCAPE BOTTOM IMMEDIATE
1210     END-IF
1220 END-REPEAT
1230 END

```

Figure 4-11 Source code listing for modified object WEBP1

**Step 15** Review the generated subprogram WEBP1N01. This object handles the processing required to represent the map WEBM1 via the Internet, based on the original presentation layer statement INPUT USING MAP 'WEBM1' found at line 0570 in program WEBP1.

The following Figure 4-12 illustrates the source code listing for the generated subprogram WEBP1N01.

```

0010 * Subprogram: WEBP1N01
0020 *****
0030 * Created by NEE on 2003-02-14 at 16:16:50.3
0040 * Created from WEBP1 from line range 0430-0439
0050 *****
0060 DEFINE DATA
0070 PARAMETER USING NEESTD-A
0080 PARAMETER
0090 01 #WEBP1N01-PDA
0100 02 #ACTION (A1/1:15)
0110 02 #MFIRST-NAME (A20/1:15)
0120 02 #MMAR-STAT (A1/1:15)
0130 02 #MMIDDLE-I (A1/1:15)
0140 02 #MNAME (A20/1:15)
0150 02 #MPERSONNEL-ID (A8/1:15)
0160 02 #MSEX (A1/1:15)
0170 02 #START (A8)
0180 02 #@NEE-SYS-LIBRARY-ID (A8)
0190 02 #@NEE-SYS-DAT4U (A10)
0200 02 #@NEE-SYS-USER (A8)
0210 02 #@NEE-SYS-PROGRAM (A8)
0220 *
0230 LOCAL USING NEESTD-L
0240 LOCAL
0250 01 #ETB
0260 02 #ETB-IO (A250/1:60)
0270 02 REDEFINE #ETB-IO
0280 03 #ETB-SB (A250/1:60)
0290 02 REDEFINE #ETB-IO
0300 03 #ETB-RB (A250/1:60)
0310 02 REDEFINE #ETB-IO
0320 03 #ETB-DATA-BLOCK
0330 04 #ACTION (A1/1:15)
0340 04 #MFIRST-NAME (A20/1:15)
0350 04 #MMAR-STAT (A1/1:15)
0360 04 #MMIDDLE-I (A1/1:15)
0370 04 #MNAME (A20/1:15)
0380 04 #MPERSONNEL-ID (A8/1:15)
0390 04 #MSEX (A1/1:15)
0400 04 #START (A8)
0410 04 #@NEE-SYS-LIBRARY-ID (A8)
0420 04 #@NEE-SYS-DAT4U (A10)
0430 04 #@NEE-SYS-USER (A8)
0440 04 #@NEE-SYS-PROGRAM (A8)
0450 02 #ETB-EB (A40)

```

```

0460 02 #ETB-EC (A08)
0470 *
0480 01 #HTML-FB (A86/1:165) INIT
0490 (1) <'#ACTION A001 00015
0500 (2) <'#MFIRST-NAME A020 00015
0510 (3) <'#MMAR-STAT A001 00015
0520 (4) <'#MMIDDLE-I A001 00015
0530 (5) <'#MNAME A020 00015
0540 (6) <'#MPERSONNEL-ID A008 00015
0550 (7) <'#MSEX A001 00015
0560 (8) <'#START A008
0570 (9) <'#@NEE-SYS-LIBRARY-ID A008
0580 (10) <'#@NEE-SYS-DAT4U A010
0590 (11) <'#@NEE-SYS-USER A008
0600 (12) <'#@NEE-SYS-PROGRAM A008
0610 *
0620 END-DEFINE
0630 IF #@NEE-STACK EQ TRUE
0640 RESET #@NEE-STACK
0650 ESCAPE ROUTINE
0660 END-IF
0670 MOVE *LIBRARY-ID TO #WEBP1N01-PDA.#@NEE-SYS-LIBRARY-ID
0680 *
0690 MOVE BY NAME #WEBP1N01-PDA TO #ETB-DATA-BLOCK
0700 MOVE 'WEBM1' TO #@NEE-MAP-NAME
0710 MOVE 'WEBM1' TO #ETB.#@NEE-SYS-PROGRAM
0720 MOVE TRUE TO #@NEE-OUTPUT-PFKEYS
0730 *
0740 CALLNAT 'NEEDAT-N' #@NEE-STD-PARMS #ETB #HTML-FB(*)
0750 *
0760 CALLNAT 'NEEETB-N' 'CONVERSE' #ETB
0770 *
0780 CALLNAT 'NEEDCP-N' #@NEE-STD-PARMS #ETB-RB(*) #HTML-FB(*)
0790 *
0800 CALLNAT 'NEECMD-N' #@NEE-STD-PARMS
0810 *
0820 RESET #@NEE-FOCUS-FIELD #@NEE-REINPUT-LINE
0830 IF #@NEE-PF-KEY EQ 'PA1' OR EQ 'PA2' OR EQ 'CLR'
0840 ESCAPE ROUTINE
0850 END-IF
0860 *
0870 MOVE #ETB.#ACTION(1:15) TO #WEBP1N01-PDA.#ACTION(1:15)
0880 MOVE #ETB.#MFIRST-NAME(1:15) TO #WEBP1N01-PDA.#MFIRST-NAME(1:15)
0890 MOVE #ETB.#MMAR-STAT(1:15) TO #WEBP1N01-PDA.#MMAR-STAT(1:15)
0900 MOVE #ETB.#MMIDDLE-I(1:15) TO #WEBP1N01-PDA.#MMIDDLE-I(1:15)
0910 MOVE #ETB.#MNAME(1:15) TO #WEBP1N01-PDA.#MNAME(1:15)
0920 MOVE #ETB.#MPERSONNEL-ID(1:15) TO #WEBP1N01-PDA.#MPERSONNEL-ID(1:15)
0930 MOVE #ETB.#MSEX(1:15) TO #WEBP1N01-PDA.#MSEX(1:15)
0940 MOVE #ETB.#START TO #WEBP1N01-PDA.#START

```

```
0950 *
0960 END
```

Figure 4-12 Source code listing for generated subprogram WEBP1N01

**Step 16** Review the generated XSL file WEBM1.XSL. This is achieved by locating the web server directory NEEEXWBX and selecting **WEBM1.XSL**.

The following Figure 4-13 illustrates sample blocks from WEBM1.XSL.

```
<td colspan="32" >
EMPLOYEE&#160;MAINTENANCE&#160;APPLICATION
</td>
<td colspan="15">&#160;</td>
<td colspan="10">
<input type="TEXT" class="outputonly" name="NEESYSDAT4U"
value="{NEESYSDAT4U}" readonly="readonly" maxlength="10" size="10"
np="rc=001070 AD=W"/>
</td>
</tr>
<tr>
<td colspan="1">&#160;</td>
<td colspan="8">
<input type="TEXT" class="outputonly" name="NEESYSUSER" value="{NEESYSUSER}"
readonly="readonly" maxlength="8" size="8" np="rc=002002 AD=W"/>
</td>
<td colspan="19">&#160;</td>
<td colspan="20" >
-&#160;BROWSE&#160;EMPLOYEES&#160;-
</td>
<td colspan="21">&#160;</td>
<td colspan="8">
<input type="TEXT" class="outputonly" name="NEESYSPROGRAM"
value="{NEESYSPROGRAM}" readonly="readonly" maxlength="8" size="8"
np="rc=002070 AD=W"/>
</td>
</tr>
<tr><td colspan="80">&#160;</td></tr>
<tr>
<td colspan="1">&#160;</td>
<td colspan="10" >
ACT&#160;EMP-ID
</td>
<td colspan="5">&#160;</td>
<td colspan="9" >
LAST-NAME
</td>
<td colspan="13">&#160;</td>
<td colspan="10" >
```

## 4

### Natural Engineer WebStar

```
FIRST-NAME
</td>
<td colspan="10">&#160;</td>
<td colspan="17" >
INIT&#160;SEX&#160;MAR-STAT
</td>
</tr>
```

Figure 4-13 Sample blocks from WEBM1.XSL

**Step 17** Add missing application objects to the modification library NEEEXWBX.

The Modification process will only create objects from an application, where the Impact Analysis process has identified presentation layer statements. Any application objects that contain no presentation layer statements will not be copied into the modification library. These will need to be copied across from the base application library.

This can be achieved by using the object maintenance utility SYSMAIN and selecting the 'copy' function. Copy the objects from library NEEEXWB to NEEEXWBX. The following objects must be omitted from the copy, as they will overwrite the Webstar modified versions in modification library NEEEXWBX:

```
WEBP1
WEBN1
```

Repeat the copy process to copy the DDM objects EMPLOYEES and VEHICLES.

There will now be a total of 15 objects in the modification library NEEEXWBX.

**Step 18** Add the WebStar initialization object. The object can be found on the supplied WebStar run time library SYSNEEW and should be copied from SYSNEEW to the modification library NEEEXWBX. The object name is **NEEINI-S** (the subroutine name is NEE-READ-INITIALISATION).

The initialization object needs to be modified to accommodate the environment settings being used.

*Note: This data is dependent on the EntireX Broker environment being used. For the purpose of this example the following is used:*

*Broker-Id = ETB255, SERVER NAME = NEEEXWBX, application start up program = WEBP1.*

Object **NEEINI-S** is modified to change the **SERVER-NAME** from the default value '???????' to 'NEEEXWBX'.

**Step 19** Configure the Natural Parameter File for the web enabled application.

A Natural Parameter module NEEEXWBX needs to be created specifying the correct work files, database files and steplib information.

The important issue is to ensure the correct steplib structure is defined so that the WebStar run time library SYSNEEW is included. For this example, the following needs to be set:

```
NEEEXWBX  
SYSNEEW
```

**Step 20** Catalog the modified application NEEEXWBX. At this stage the modified application library is complete. The whole library is cataloged ready for execution.

The catalog results will show 16 objects cataloged with no errors or failures.

*Note: At this point it is possible to execute the modified application NEEEXWBX, using a Natural session and invoking the application using the start up object WEBP1. The modified application will execute as a normal Natural application, in fact the same as before the WebStar development function was applied.*

**Step 21** Copy the template page 'TEMPL.HTML' from the NEE web server directory to the application web server directory NEEEXWBX. This page is used for the left hand frame.

**Step 22** EntireX Broker. A Broker Service needs to be created which will be used to handle the communication between the web server and Natural.

For the purpose of this example a Broker ETB255 is used with the following server-specific Broker Attribute File settings:

```
CLASS = NEE, SERVER = NEEEXWBX, SERVICE = INPUT  
CLASS = NEE, SERVER = NEEWEB, SERVICE = NEEWEB  
CLASS = NEE, SERVER = NEEATM, SERVICE = END
```

## 4

### Natural Engineer WebStar

**Step 23** Configure the Attach Manager initialization file NEEATM.INI.

The Attach Manager program NEEATM.EXE, is used to initiate and maintain the Natural sessions for the web enabled application. The initialization file for this example has been specified as follows:

```
APP-DEF
APP-TITLE=WebStar Sample Application (NEEEXWBX)
APP-BROKER-ID=ETB255
APP-SERVER-CLASS=NEE
APP-SERVER-NAME=NEEEXWBX
APP-SERVICE=INPUT
APP-EXEC=D:\Program Files\Software AG\Natural\5.1.1\bin\natural.exe
APP-PARM=PARM=NEEEXWBX STACK=(LOGON NEEEXWBX;WEB:WEBP1)
```

**Step 24** Configure the web server interface initialization file NEEC.INI.

The web server initialization file NEEC.INI, is used to initiate the parameter settings required to map to the correct EntireX Broker service. The initialization file for this example has been specified as follows:

```
HTML_TEMPLATE_DIR=D:\inetpub\wwwroot\nee
APP_FILE=
D:\Program Files\Software AG\Natural Engineer\4.4.2\web\neeatm\ neeatm.ini
TEMP_DIR=D:\Program Files\Software AG\Natural Engineer\4.4.2\web
```

*Note: Only the modified parameters for this example are shown. All others are as per default.*

**Step 25** Start the SAG demo database: SAG-DEMO-DB.

The SAG demo database contains the EMPLOYEES and VEHICLES files used by the application NEEEXWBX.

The database can be started from DBA Workbench:

1. Select database SAG-DEMO-DB
2. Select drop menu option Database → Start.

**Step 26** Executing the web enabled application over the Internet.

1. Start the Broker session. For this example Broker ETB255 is being used.

The following Figure 4-14 illustrates the EntireX Broker ETB255 started.

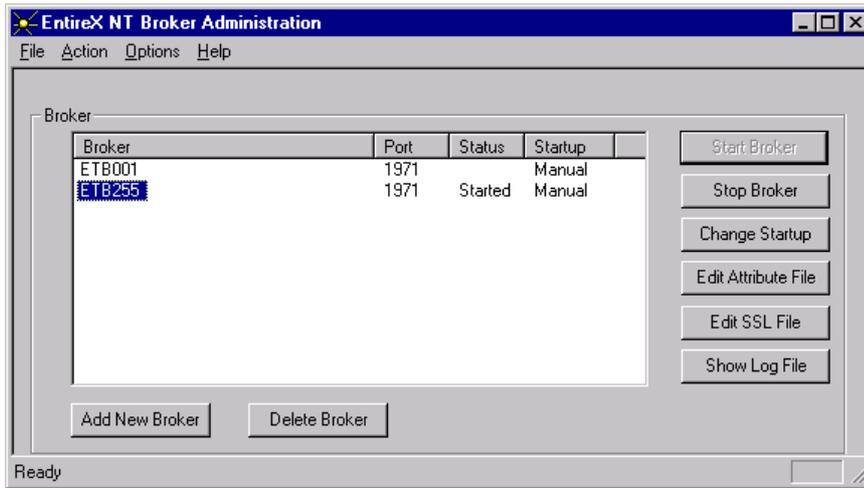


Figure 4-14 EntireX Broker ETB255 started

2. Start the Attach Manager NEEATM.

The Attach Manager can be started using the Start→Run command window and typing in:

X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\WEB\NEEATM\NEEATM.EXE

(where X: is the directory on which Natural Engineer was installed).

The following Figure 4-15 illustrates the Attach Manager NEEATM start screen.

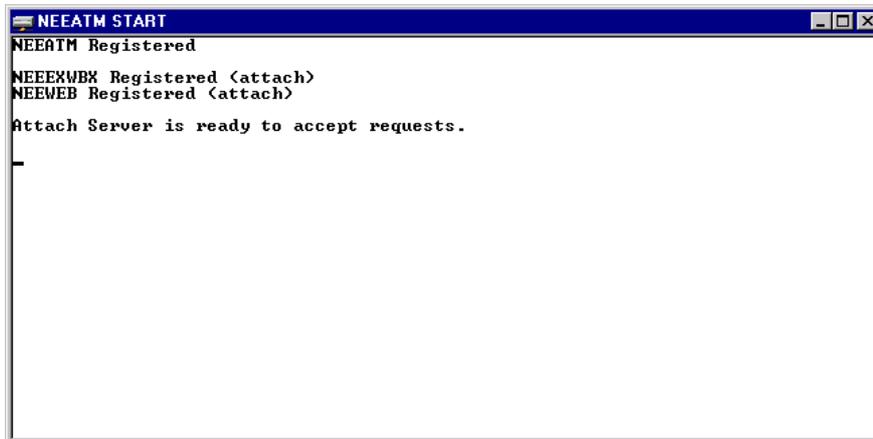


Figure 4-15 Attach Manager NEEATM start screen

## 4

### Natural Engineer WebStar

3. Open the web browser and type in the start up URL: <http://localhost/nee/webstar.html>.

*Note: 'localhost' will need to be the correct IP address of the web server you are using.*

The following Figure 4-16 illustrates the web browser with the start up URL.

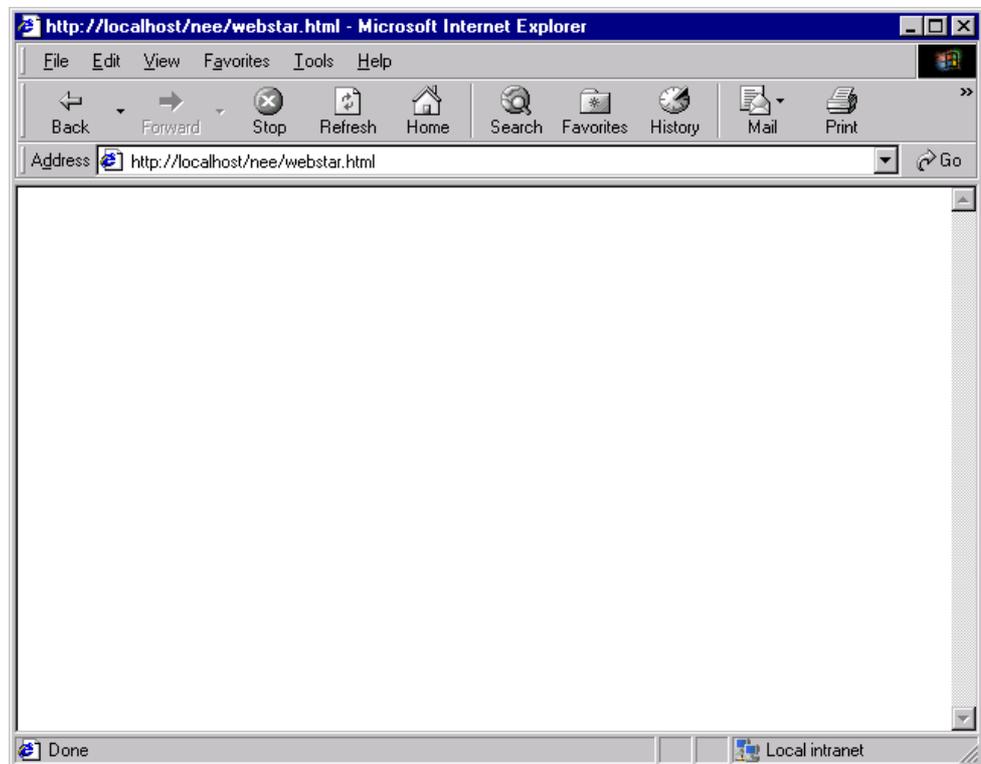


Figure 4-16 Web Browser with the start up URL

4. The WebStar Logon page will be displayed. Here the User-id and password are input, and the application selected.

*Note: The normal User-id and password for the application are entered as per site standards. These can be validated using the EntireX Security functionality.*

For this example use the following:

USERID = NEEWEB  
PASSWORD = NEEWEB

The following Figure 4-17 illustrates the WebStar Logon page with details entered.

The screenshot shows a web browser window titled "Webstar - Microsoft Internet Explorer". The page content is on a yellow background with a blue sidebar on the left containing the "GSL" logo. The main heading is "WEBSTAR Logon". Below the heading are three input fields: "USERID" containing "NEEWEB", "PASSWORD" containing "NEEWEB", and "APPLICATIONs" containing "WebStar Sample Application (NEEEXWBX)". An "OK" button is located at the bottom center of the form area.

Figure 4-17 The WebStar Logon page with details entered

# 4

## Natural Engineer WebStar

5. The application NEEEXWBX is now executing via the Internet. The first page of the system: Browse Employees is displayed.

The following Figure 4-18 illustrates the application NEEEXWBX executing via the Internet.

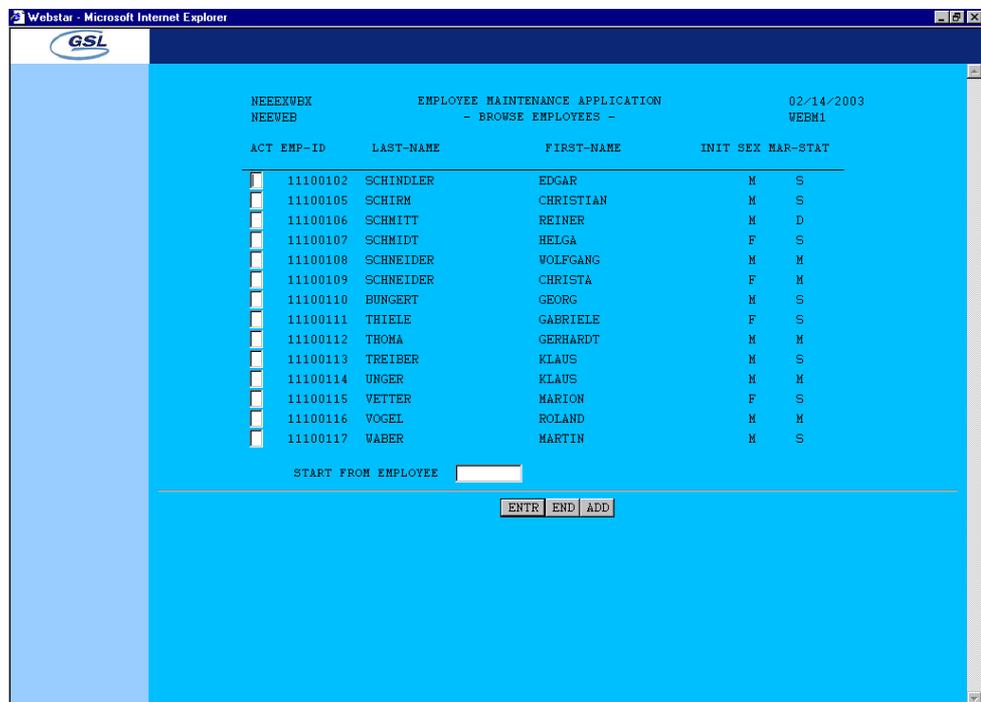


Figure 4-18 The application NEEEXWBX executing via the Internet

The Attach Manager NEEATM is now waiting for the next transaction to be passed to the Natural session, in order to process the selected option.

The application can be terminated by selecting the 'END' button. This will close the browser window and the Natural Session. The Broker ETB255 and the Attach Manager will need to be manually closed down.

# SHORT TRANSACTION PROCESSING

## Chapter Overview

---

This chapter explains the processes required for Short Transaction Processing using the Natural Engineer add-on component WebStar.

The topics covered are:

- [Introduction](#)
- [Creating Short Transactions](#)
- [Example Short Transactions](#)

## Introduction

---

The Natural Engineer add-on component WebStar run time function provides the facility to run Natural requests via the Internet. This is collectively known as Short Transaction Processing.

Short Transactions are when a request is sent to the Natural Server, the Natural Server services the transaction and returns the results back to the web browser. The Short Transaction is now over.

Some examples of Short Transactions:

- List objects within a Natural library.
- List source code for a selected object.
- Read a database file and display record data.

Data from a Short Transaction can be received in several different formats providing flexibility in reviewing the requested data and saving the data in different formats. The supported formats are:

- Standard XSL
- Standard HTML.
- Adobe Portable Document Format (PDF).
- Microsoft Excel.
- Microsoft Word.
- XML.

*Note: WebStar Short Transaction processing has no data size limitations. The data for each Short Transaction is returned until all data has been retrieved.*

## Creating Short Transactions

---

### Overview

The WebStar Short Transaction Processing combines components in the Natural, Web Server and WebStar run time environments to be generated and/or maintained in order for Natural Short Transactions to be executed via the Internet.

Natural objects within a Natural library are manually created for each Short Transaction required, utilizing the Short Transaction Exit routines supplied in the WebStar run time library SYSNEEW. These will contain the processing logic for both the Natural Short Transaction itself and the type of presentation to be used.

Web page templates can be created to present the Natural Short Transaction results data in the web browser. These will contain any required page styles and data references used to merge the data from the Natural object with the web page template. XML data can be returned for processing within a calling application.

The EntireX Broker Instance being used for Natural Short Transactions must be invoked using the EntireX Broker Administration function and selecting the Broker to be used, for example:

```
ETB255
```

*Note: The EntireX Broker Attribute File for the Broker Instance needs to be checked to ensure that it contains a service identity for Natural Short Transactions, using a service and server of NEEWEB. For example:*

```
CLASS = NEE, SERVER = NEEWEB, SERVICE = NEEWEB
```

An Attach Manager must be invoked to establish the correct Natural server session information to execute the Natural Short Transaction. This can be the EntireX Attach Service if using a mainframe environment, or the WebStar Attach Manager program NEEATM.EXE if using a Windows environment.

## 5

### Natural Engineer WebStar

The Natural Short Transaction can then be executed by inputting the relevant URL in the web browser address line. If the Natural Short Transaction requires any input data parameters, these can be specified as strings appended to the URL address.

Examples:

- 1) `http://www.myweb.com/scripts/neec.exe/neeexst/myquery`
- 2) `http://www.myweb.com/scripts/neec.exe/neeexst/natdir?LIB=TEST`
- 3) `http://www.myweb.com/scripts/neec.exe/neeexst/natsrc?LIB=TEST&OBJ=PROG1`

This will utilize a Natural server session that may already be available or initiate a new session.

The web server ('www.myweb.com' in the URL example 1) is accessed and the web interface program NEEC.EXE will control the communication between the web server and EntireX Broker.

The Natural library ('neeexst' in the URL example 1) containing the Natural Short Transaction object to be executed.

The Natural Short Transaction object ('myquery' in the URL example 1) will then execute, formatting the data in XML and sending it back to the web browser, where it will be displayed.

The Short Transaction has now completed and the Natural server is now available to process another Short Transaction or terminate if the timeout parameter is reached.

## Coding Short Transactions

Natural objects containing the required Short Transaction logic need to be manually created on a Natural library. The Natural objects can be saved as either program or subprogram object types.

Each Natural Short Transaction object will contain:

- User processing logic to provide the Short Transaction results data
- Reference to Short Transaction Exit routines supplied on the WebStar run time library SYSNEEW. These will correctly handle the format and presentation characteristics for the requested data between the Natural server session and the web server.

*Note: There are no data limitations when running Short Transactions. All output data is sent to the web browser until all output data has been exhausted.*

The Short Transaction Exit routines required are:

- [Data Buffer Area](#)
- [Initialize Data Buffer Area](#)
- [Read Input Data Parameter Strings](#)
- [Read Data Buffer Area](#)
- [Initialize Short Transaction Output](#)
- [Change Content Type](#)
- [Output Short Transaction Data using Data Substitution](#)
- [Output Short Transaction Data using Free Format](#)
- [End Short Transaction](#)

Each of these are now described.

## Data Buffer Area

Each Natural Short Transaction object created needs to reference the Data Buffer Area used to pass the Short Transaction results data between the Natural server session and the web server via EntireX Broker.

The Data Buffer Area is referenced by using the variable #ETB. The definition for this variable is contained in the Parameter Data Area NEEETB-A.

This should be included at the start of the Data Definition section.

The use of this definition is mandatory in all Natural Short Transaction objects.

Example:

```
0110 DEFINE DATA
0120 LOCAL USING NEEETB-A
0130 LOCAL
0140 01 EMP-FILE VIEW EMPLOYEES
0150   02 PERSONNEL-ID
0160   02 FIRST-NAME
0170   02 NAME
0180   02 BIRTH
0190 01 #COUNT (I02) INIT <+0>
0200 END-DEFINE
```

## Initialize Data Buffer Area

The Data Buffer Area (#ETB) needs to be initialized at the start of the Natural Short Transaction.

This is done using the following statement:

```
PERFORM WS-INIT #ETB
```

Mandatory parameters:

```
#ETB      Data buffer area used to pass the Short Transaction results data between the  
           Natural server session and the web server via EntireX Broker.
```

This must be the first executable statement in the Natural Short Transaction object.

Example:

```
0200 END-DEFINE  
0210 /*  
0220 /* INITIALIZE ETB BUFFER  
0230 PERFORM WS-INIT #ETB
```

## Read Input Data Parameter Strings

If the Natural Short Transaction is to utilize any input data parameter strings from the URL address line, then the parameters need to be read in and processed.

The input data parameters are processed in pairs. Each pair consists of a parameter name and parameter value.

This is done using the following statement:

```
PERFORM WS-READ-ENVIRONMENT #ETB operand1 operand2 operand3
```

Mandatory parameters:

<b>#ETB</b>	Data buffer area used to pass the Short Transaction results data between the Natural server session and the web server via EntireX Broker.
<b>operand1</b>	The input data parameter name. This must contain the name of the parameter for which the value is required.  Can be a literal string or a user defined alphanumeric variable of length 1 to 250.
<b>operand2</b>	The input data parameter value. This is returned by the performed routine.  Must be defined as an alphanumeric variable of length 1 to 250.
<b>operand3</b>	A length variable defined as I04. The length value is returned by the performed routine.

This needs to be repeated for each parameter pairing used in the URL address. If no input data parameters are used then this statement is not required.

The handling of input data parameter strings would typically be done at the top of the Natural Start Transaction, after the initialization of the Data Buffer Area.

Example 1:

```
http://www.myweb.com/scripts/nec.exe/necxst/natdir?LIB=TEST
```

This URL address contains one set of parameter pairs, the parameter name is 'LIB' and the parameter value is 'TEST'. This would need the following statement:

```
0100 PERFORM WS-READ-ENVIRONMENT #ETB 'LIB' #VALUE #LENGTH
```

This perform would return the value 'TEST' which would be stored in #VALUE. This can now be referenced within the Natural Short Transaction object.

Example 2:

**http://www.myweb.com/scripts/nec.exe/necxst/natsrc?LIB=TEST&OBJ=PGM1**

This URL address contains two sets of parameter pairs, the first parameter name is 'LIB' and the first parameter value is 'TEST', the second parameter name is 'OBJ' and the second parameter value is 'PROG1', This would need the following statements:

```
0100 PERFORM WS-READ-ENVIRONMENT #ETB 'LIB' #VALUE1 #LENGTH
0110 PERFORM WS-READ-ENVIRONMENT #ETB 'OBJ' #VALUE2 #LENGTH
```

The first perform would return the value 'TEST' which would be stored in #VALUE1 and the second perform would return the value 'PGM1' which would be stored in #VALUE2. Both #VALUE1 and #VALUE2 can now be referenced within the Natural Short Transaction object.

*Note: Using PERFORM WS-READ-ENVIRONMENT as described above, is restricted to a maximum data length of 250 bytes for all data being returned (operand2). This limitation can be resolved by using the statement:*

```
PERFORM WS-READ-ENVIRONMENTA #ETB operand1 operand2 operand3
```

*Operand2 would need to be defined as a DYNAMIC variable. For example:*

```
01 #VALUE (A) DYNAMIC
```

*This permits data of any length to be returned by operand2. Use of this alternate statement is restricted to Natural running under the Windows operating system.*

## Read Data Buffer Area

The contents of the Data Buffer Area (#ETB) can be read in and its contents are then available for use within the Natural Short Transaction.

This is done using the following statement:

```
PERFORM WS-READ-TEXT #ETB operand1 operand2
```

Mandatory parameters:

<b>#ETB</b>	Data buffer area used to pass the Short Transaction results data between the Natural server session and the web server via EntireX Broker.
<b>operand1</b>	The variable which will contain the Data Buffer Area contents returned from the routine.  It is defined as a DYNAMIC variable. For example: 01 #TEXT-STRING (A) DYNAMIC
<b>Operand2</b>	A length variable defined as I04. The length value is returned by the performed routine.

This would typically be done at any point within the Natural Short Transaction where the data in the Data Buffer Area is required.

The contents of the Data Buffer Area need to be read in before any WRITE processing, as the Data Buffer Area is shared by both the READ and WRITE Natural Short Transaction exit routines.

*Note: Use of this statement is restricted to Natural running under the Windows operating system.*

Example:

```
0230 PERFORM WS-READ-TEXT #ETB #TEXT-STRING #LENGTH
```

The contents of the Data Buffer Area are returned in variable #TEXT-STRING. This can then be manipulated within the Natural Short Transaction.

## Initialize Short Transaction Output

All Short Transaction output needs to be initialized before any output is started, and then after all output has been completed.

This is done using the following statement:

```
PERFORM WS-WRITE-SHORT-INIT #ETB operand1 operand2
```

Mandatory parameters:

<b>#ETB</b>	Data buffer area used to pass the Short Transaction results data between the Natural server session and the web server via EntireX Broker.
<b>operand1</b>	<p>The file name of the web page template to be used. The name should not include the file extension. For example:</p> <p>If the template '<b>natemp.html</b>' were to be used, operand1 would be set to '<b>natemp</b>'.</p> <p>Can be a literal string or a user defined alphanumeric variable of length 1 to 32.</p> <p><i>Note: Operand1 can be blank if no predefined web page templates are available and the result data is formatted using the Free Format output method.</i></p>
<b>operand2</b>	<p>Indicates start or end of a set of data. Possible values are:</p> <p><b>'I'</b> Initialization at the start of data output. Mandatory for both Data Substitution and Free Format output methods.</p> <p><b>''</b> Initialization at the end of data output. Mandatory for Data Substitution output method. Optional for Free Format output method.</p> <p>Can be a literal string or a user defined alphanumeric variable of length 1 to 8.</p> <p><i>Note: For more information on the Data Substitution output method refer to section <a href="#">Output Short Transaction Data using Data Substitution</a>. For more information on the Free Format output method refer to section <a href="#">Output Short Transaction Data using Free Format</a>.</i></p>

## 5

### Natural Engineer WebStar

Example 1:

The following example code shows the Short Transaction output initialization for a Natural Short Transaction that will read 10 records from the EMPLOYEES database file and output the results for the database fields PERSONNEL-ID, FIRST-NAME, NAME and BIRTH.

This example is using the Data Substitution output method, with the results data output to an HTML page template.

```
0250 /* INITIATE THE START OF SHORT TRANSACTION OUTPUT
0260 /* USING TEMPLATE NATEMP.HTML/NATEMP.XSL
0270 PERFORM WS-WRITE-SHORT-INIT #ETB 'NATEMP' 'I'
0280 /*
0290 /* READ 10 EMPLOYEE RECORDS AND DISPLAY INFORMATION
0300 READ (10) EMP-FILE
0310 /* INITIALIZE THE HTML NEE TAG
0320 PERFORM WS-WRITE-ENVIRONMENT #ETB 'HTML-NEE-TAG' 'NEE_LINE'
0330 /* OUTPUT THE DATA NAME & VALUE PAIRS
0340 PERFORM WS-WRITE-ENVIRONMENT #ETB 'ID' EMP-FILE.PERSONNEL-ID
0350 PERFORM WS-WRITE-ENVIRONMENT
0360 #ETB 'FIRST-NAME' EMP-FILE.FIRST-NAME
0370 PERFORM WS-WRITE-ENVIRONMENT #ETB 'NAME' EMP-FILE.NAME
0380 PERFORM WS-WRITE-ENVIRONMENT #ETB 'BIRTH' EMP-FILE.BIRTH
0390 ADD 1 TO #COUNT
0400 END-READ
0410 /*
0420 /* INITIALIZE THE END OF SHORT TRANSACTION OUTPUT
0430 PERFORM WS-WRITE-SHORT-INIT #ETB 'NATEMP' ' '
```

The Perform statement at line **0270** initializes the start of the Short Transaction output, passing the file name 'NATEMP' for the HTML page template 'natemp.html' and 'I' in operand2.

The Perform statement at line **0430** initializes the end of the Short Transaction output, passing the file name 'NATEMP' for the HTML page template 'natemp.html' and ' ' (blank) in operand2.

Example 2:

The following example code shows the Short Transaction output initialization for a Natural Short Transaction that will read records from the VEHICLES database file and output the results for the database fields MAKE and MODEL.

This example is using the Free Format output method with the results data and HTML tags formatted in-line.

```

0240 /* INITIATE THE START OF SHORT TRANSACTION OUTPUT
0250 PERFORM WS-WRITE-SHORT-INIT #ETB ' ' 'I'
0260 /*
0270 PERFORM WS-WRITE-TEXT #ETB '<table>'
0280 PERFORM WS-WRITE-TEXT #ETB
0290 ' <tr><th>Make</th><th>Model</th></tr>'
0300 READ VEHICLES
0310 PERFORM WS-WRITE-TEXT #ETB '<tr>'
0320 PERFORM WS-WRITE-TEXT #ETB '<td>'
0330 PERFORM WS-WRITE-TEXT #ETB VEHICLES.MAKE
0340 PERFORM WS-WRITE-TEXT #ETB '</td>'
0350 PERFORM WS-WRITE-TEXT #ETB '<td>'
0360 PERFORM WS-WRITE-TEXT #ETB VEHICLES.MODEL
0370 PERFORM WS-WRITE-TEXT #ETB '</td>'
0380 PERFORM WS-WRITE-TEXT #ETB '</tr>'
0390 END-READ
0400 PERFORM WS-WRITE-TEXT #ETB '</table>'
0410 /*
0420 /* CLOSE ETB BUFFER
0430 PERFORM WS-END #ETB
0440 /*
0450 END

```

The Perform statement at line **0250** initializes the start of the Short Transaction output, passing a blank file name '' for the web page template name (as no template is being used) and 'I' in operand2.

There is no statement to initialize the end of the Short Transaction output, as it is optional for the Free Format output method. It could be included at line 0410 as:

```
0410 PERFORM WS-WRITE-SHORT-INIT #ETB ' ' ' ' ' '
```

The result would be the same.

## Change Content Type

If a Short Transaction result is required in a non-HTML/XSL format, for example in Microsoft Excel format, then the content type (also known as MIME type) can be changed.

This is done using the following statement:

**PERFORM WS-WRITE-CONTENT-TYPE #ETB operand1**

Mandatory parameters:

**#ETB** Data buffer area used to pass the Short Transaction results data between the Natural server session and the web server via EntireX Broker.

**operand1** The Content type (also known as MIME type) to be used.

Can be a literal string or a user defined alphanumeric variable of length 1 to 250.

This statement should immediately follow the initialization of the start of Short Transaction output (WS-WRITE-SHORT-INIT).

This routine is only required if an alternate Content type is required.

Example 1:

```
0290 PERFORM WS-WRITE-SHORT-INIT #ETB ' ' 'I'
0300 PERFORM WS-WRITE-CONTENT-TYPE #ETB 'application/msword'
```

This will tell the web browser that the data is to be presented as a Microsoft Word document.

Example 2:

```
0290 PERFORM WS-WRITE-SHORT-INIT #ETB ' ' 'I'
0300 PERFORM WS-WRITE-CONTENT-TYPE #ETB 'application/vnd.ms-excel'
```

This will tell the web browser that the data is to be presented as a Microsoft Excel document.

## Output Short Transaction Data using Data Substitution

Any Short Transaction results that are to be displayed using web page templates are output using 'data name pairs'. The data name pairs consist of a data name and a data value. This method is known Data Substitution.

This is done using the following statement:

```
PERFORM WS-WRITE-ENVIRONMENT #ETB operand1 operand2
```

Mandatory parameters:

<b>#ETB</b>	Data buffer area used to pass the Short Transaction results data between the Natural server session and the web server via EntireX Broker.
<b>operand1</b>	The data name used within the web page templates. The data name must match the data name that has been defined in the web page template. It does not need to be the same as the variable name containing the data.  Can be a literal string a database field or a user defined alphanumeric variable of length 1 to 250.
<b>operand2</b>	The data value to be inserted in the web page templates.  Can be a literal string or a user defined alphanumeric variable of length 1 to 250.

The output process needs to know where the data name pairs are located in the web page templates. This is handled by sending a data name pair containing a dummy data name of 'HTML-NEE-TAG' and a data value, example '<NEE\_LINE>' at the start of output processing.

*Note: The data name 'HTML-NEE-TAG' is used by the internal Short Transaction process to locate the data names within the web page templates, and is mandatory. The data value '<NEE\_LINE>' can be any value required, and must match the tag used within the web page template.*

This is done using the following statement:

```
PERFORM WS-WRITE-ENVIRONMENT #ETB 'HTML-NEE-TAG' 'NEE_LINE'
```

This statement should always precede any other data name pair statements being used:

```
PERFORM WS-WRITE-ENVIRONMENT #ETB 'HTML-NEE-TAG' 'NEE_LINE'  
PERFORM WS-WRITE-ENVIRONMENT #ETB 'data-name' 'data value'
```

# 5

## Natural Engineer WebStar

Example:

HTML code:

```
<NEE>
  <NEE_LINE>
    <tr>
      <td>$ID$</td>
      <td>$NAME$</td>
    </tr>
  </NEE_LINE>
</NEE>
```

XSL code:

```
<xsl:for-each select="NEE">
  <xsl:for-each select="NEE_LINE">
    <tr>
      <td><xsl:value-of select="ID" /></td>
      <td><xsl:value-of select="NAME" /></td>
    </tr>
  </xsl:for-each>
</xsl:for-each>
```

Natural code:

```
0280 READ EMPLOYEES
0290   PERFORM WS-WRITE-ENVIRONMENT #ETB 'HTML-NEE-TAG' 'NEE_LINE'
0300   PERFORM WS-WRITE-ENVIRONMENT #ETB 'ID' EMP-FILE.PERSONNEL-ID
0310   PERFORM WS-WRITE-ENVIRONMENT #ETB 'NAME' EMP-FILE.NAME
0320 END-READ
```

This example is reading the EMPLOYEES database file and displaying the database fields PERSONNEL-ID and NAME. The read logic contains the Short Transaction output statements sending the data name pairs to a web page template, with the first output statement at line 0290 sending the location of the data identities within the web page template.

## Output Short Transaction Data using Free Format

Short Transaction results can also be formatted within the Natural Short Transaction object when not using a predefined web page template. This is known as Free Format.

This is done using the following statement:

```
PERFORM WS-WRITE-TEXT #ETB operand1
```

Mandatory parameters:

<b>#ETB</b>	Data buffer area used to pass the Short Transaction results data between the Natural server session and the web server via EntireX Broker.
<b>operand1</b>	Contains the required output data. Can be a literal string or a user defined alphanumeric variable of length 1 to 250.

The output data can be passed as a literal string or a user defined variable. The literal string can be a combination of literal strings and data values.

This method allows you to specify the presentation format of your data as required. This could be using HTML/XSL tags or comma delimited data for use in Microsoft Excel or Microsoft Word.

Example 1:

```
0270 PERFORM WS-WRITE-TEXT #ETB '<table>'
0280 PERFORM WS-WRITE-TEXT #ETB
0290   '<tr><th>Make</th><th>Model</th></tr>'
0300 READ VEHICLES
0310   PERFORM WS-WRITE-TEXT #ETB '<tr>'
0320   PERFORM WS-WRITE-TEXT #ETB '<td>'
0330   PERFORM WS-WRITE-TEXT #ETB VEHICLES.MAKE
0340   PERFORM WS-WRITE-TEXT #ETB '</td>'
0350   PERFORM WS-WRITE-TEXT #ETB '<td>'
0360   PERFORM WS-WRITE-TEXT #ETB VEHICLES.MODEL
0370   PERFORM WS-WRITE-TEXT #ETB '</td>'
0380   PERFORM WS-WRITE-TEXT #ETB '</tr>'
0390 END-READ
0400 PERFORM WS-WRITE-TEXT #ETB '</table>'
```

## 5

## Natural Engineer WebStar

This example is reading the VEHICLES database file and displaying the database fields MAKE and MODEL. The HTML tags used will formulate a simple two-column table. The number of rows will be determined by the number of records read.

Example 2:

```
0270 COMPRESS '<table><tr><th>Make</th><th>Model</th></tr>'
0280 INTO #HTML-LINE
0290 PERFORM WS-WRITE-TEXT #ETB #HTML-LINE
0300 READ VEHICLES
0310 COMPRESS '<tr><td>' VEHICLES.MAKE '</td>'
0320 INTO #HTML-LINE LEAVING NO SPACE
0330 PERFORM WS-WRITE-TEXT #ETB #HTML-LINE
0340 COMPRESS '<td>' VEHICLES.MODEL '</td>'
0350 INTO #HTML-LINE LEAVING NO SPACE
0360 PERFORM WS-WRITE-TEXT #ETB #HTML-LINE
0370 END-READ
0380 COMPRESS '</tr></table>' INTO #HTML-LINE LEAVING NO SPACE
0390 PERFORM WS-WRITE-TEXT #ETB #HTML-LINE
```

This example is the same as the Example 1, but this time is making use of a user defined variable #HTML-LINE to format each output line.

Example 3:

```
0310 PERFORM WS-WRITE-CONTENT-TYPE #ETB 'application/msword'
0320 COMPRESS 'Personnel Number,Name,Age,Family Status'
0330 INTO #HTML-LINE
0340 PERFORM WS-WRITE-TEXT #ETB #HTML-LINE
0350 READ PERSONNEL BY PERSONNEL-NUMBER STARTING FROM 100020
0360 COMPRESS PER-FILE.PERSONNEL-NUMBER ',' PER-FILE.NAME ','
0370 PER-FILE.AGE ',' PER-FILE.FAMILY-STATUS
0380 INTO #HTML-LINE LEAVING NO SPACE
0390 PERFORM WS-WRITE-TEXT #ETB #HTML-LINE
0400 END-READ
```

This example is reading the PERSONNEL database file and displaying the database fields PERSONNEL-NUMBER, NAME, AGE and FAMILY-STATUS. The Content type has been set at line 0330 to use Microsoft Word and the data is being output as comma delimited data.

## End Short Transaction

At the end of the Short Transaction processing the conversation between the Natural server session and EntireX Broker needs to be terminated.

This is done using the following statement:

**PERFORM WS-END #ETB**

Mandatory parameters:

**#ETB** Data buffer area used to pass the Short Transaction results data between the Natural server session and the web server via EntireX Broker.

This routine signifies the end of a conversation between the Natural server session and EntireX Broker and will send any final data.

This must be the final executable statement in the Natural Short Transaction object.

Example:

```
0480 /* CLOSE ETB BUFFER
0490 PERFORM WS-END #ETB
0500 /*
0510 END
```

## Supplied Example Short Transaction Objects

The Natural Engineer add-on component WebStar comes supplied with a Natural library NEEEXST containing a set of example Natural Short Transaction objects.

*Note: The supplied example Short Transaction objects make use of the supplied example web page templates. For more information refer to section [Supplied Example Web Page Templates](#).*

The Natural library NEEEXST contains the following objects:

Object	Description
NATDIR	<p>Program object that produces a list of objects within a Natural library.</p> <p>The output is formatted using the supplied web page templates: <a href="#">'natdir.html'</a> / <a href="#">'natdir.xml'</a>.</p> <p>From this list, individual objects can be selected to list the object source code using the example Natural Short Transaction object <a href="#">NATSRC</a>.</p> <p>The library name can be entered using the input data parameter string: <b>'?LIB=XXXXXXXX'</b>, where:</p> <p><b>?</b> Indicates the start of the input data parameter.</p> <p><b>LIB=</b> The name of the data parameter.</p> <p><b>XXXXXXXX</b> The value of the data parameter. This will be the name of the Natural library.</p> <p><i>Note: If the input data parameter is not specified, you will get a list of the objects in the default logon library for the Natural server session being used.</i></p> <p>Execution examples:</p> <p><b>http://www.myweb.com/scripts/nec.exe/necxst/natdir</b></p> <p>Results in a list of all the objects in the default logon library, for example SYSTEM.</p> <p><b>http://www.myweb.com/scripts/nec.exe/necxst/natdir?LIB=TEMP</b></p> <p>Results in a list of all the objects in library TEMP.</p>

<b>Object</b>	<b>Description</b>
<b>NATEMP</b>	<p>Program object that will read 10 records from the EMPLOYEES database file and display the record details in a table.</p> <p>The output is formatted using the supplied web page templates: <a href="#">'natemp.html'</a> / <a href="#">'natemp.xml'</a>.</p> <p>There are no input data parameters required for this object.</p> <p>Execution example:</p> <p><b><a href="http://www.myweb.com/scripts/nec.exe/necxst/natemp">http://www.myweb.com/scripts/nec.exe/necxst/natemp</a></b></p>
<b>NATEMPN</b>	<p>Subprogram object that will read 100 records from the EMPLOYEES database file and display the record details in a table.</p> <p>The output is formatted using the supplied web page templates: <a href="#">'natemp.html'</a> / <a href="#">'natemp.xml'</a>.</p> <p>There are no input data parameters required for this object.</p> <p>Execution example:</p> <p><b><a href="http://www.myweb.com/scripts/nec.exe/necxst/natempn">http://www.myweb.com/scripts/nec.exe/necxst/natempn</a></b></p>
<b>NATEXCEL</b>	<p>Program object that will read 10 records from the PERSONNEL file and display the record details in a table using a MIME type of <b>'application/vnd.ms-excel'</b>.</p> <p>The output will be formatted comma-delimited text, which will be displayed in the web browser. The Microsoft Excel program will be opened within the web browser so that the results can be saved as an Excel document.</p> <p><i>Note: The table data can be formatted using the 'convert text to columns' option within Excel.</i></p> <p>There are no parameters required for this object.</p> <p>Execution example:</p> <p><b><a href="http://www.myweb.com/scripts/nec.exe/necxst/natexcel">http://www.myweb.com/scripts/nec.exe/necxst/natexcel</a></b></p>

## 5

## Natural Engineer WebStar

Object	Description
NATSRC	<p>Program object that will produce a source code listing for a specified / selected object.</p> <p>The output is formatted using the supplied web page templates: <a href="#">'natsrc.html'</a> / <a href="#">'natsrc.xml'</a>.</p> <p>This object requires input data parameters of a library and object name.</p> <p>The library and object names can be entered using the input data parameter string: '?LIB=XXXXXXXX&amp;OBJ=YYYYYYYY', where:</p> <p><b>?</b> Indicates the start of the first data parameter.  <b>LIB=</b> The name of the first data parameter  <b>XXXXXXXX</b> The value of the first data parameter. This will be the name of the Natural library.  <b>&amp;</b> Indicates the start of a second data parameter.  <b>OBJ=</b> The name of the second data parameter.  <b>YYYYYYYY</b> The value of the second data parameter. This will be the name of the object.</p> <p>This object can also be invoked by selecting an object from the list of objects produced by the example Natural Short Transaction object <a href="#">NATDIR</a>.</p> <p><i>Note: For this example Natural Short Transaction the input data parameter is mandatory.</i></p> <p>Execution example:</p> <p><b><a href="http://www.myweb.com/scripts/nec.exe/necxst/natsrc?LIB=TEMP&amp;OBJ=PROG1">http://www.myweb.com/scripts/nec.exe/necxst/natsrc?LIB=TEMP&amp;OBJ=PROG1</a></b></p> <p>Results in a source code listing of object PROG1 found in library TEMP.</p>
NATVEH	<p>Program object that will read 10 records from the VEHICLES database file and display the record details in a table.</p> <p>The output is formatted using in-stream HTML tags, which will be displayed in the web browser. This is an example of coding a complete HTML page within a Natural object.</p> <p>There are no input data parameters required for this object.</p> <p>Execution example:</p> <p><b><a href="http://www.myweb.com/scripts/nec.exe/necxst/natveh">http://www.myweb.com/scripts/nec.exe/necxst/natveh</a></b></p>

Object	Description
<b>NATWORD</b>	<p data-bbox="491 663 1232 748">Program object that will read 10 records from the PERSONNEL file and display the record details in a table using a MIME type of '<b>application/msword</b>'.</p> <p data-bbox="491 763 1232 848">The output will be formatted comma-delimited text, which will be displayed in the web browser. The Microsoft Word program will be opened within the web browser so that the results can be saved as a Word document.</p> <p data-bbox="491 864 1232 918"><i>Note: The table data can be formatted using the convert text to table option within Word.</i></p> <p data-bbox="491 934 959 965">There are no parameters required for this object.</p> <p data-bbox="491 981 687 1012">Execution example:</p> <p data-bbox="491 1028 1082 1050"><b><a href="http://www.myweb.com/scripts/nec.exe/necxst/natword">http://www.myweb.com/scripts/nec.exe/necxst/natword</a></b></p>

## Short Transaction Error Processing

Short Transaction error processing is provided by the WebStar run time environment. This will report any errors encountered during the execution of a Short Transaction.

When an error occurs, an error report is produced using a web page template 'naterr.html' or 'naterr.xml'.

*Note: The template error report pages 'naterr.html' and 'naterr.xml' are located in the SYSTEM folder under the web server NEE folder. For more information refer to Chapter 2 Installation and Chapter 3 Configuration.*

The following Figure 5-1 illustrates the error report when an invalid Natural Short Transaction object name has been input in the URL address line.

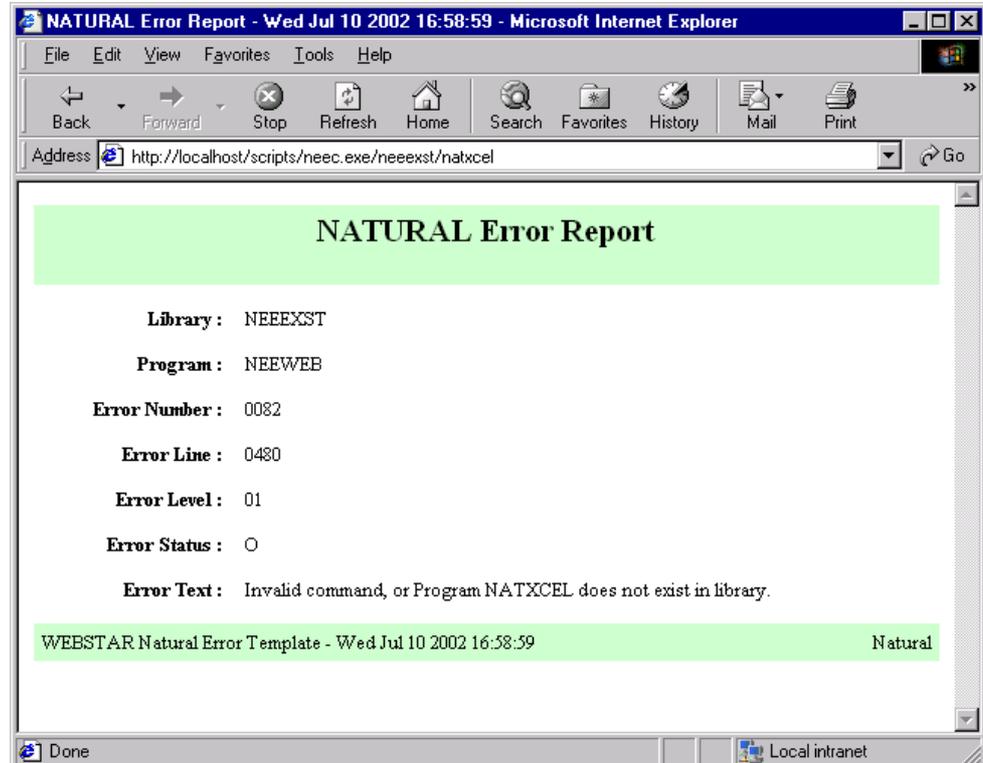


Figure 5-1 Error Report for invalid Natural Short Transaction object

The following Figure 5-2 illustrates the error report when a Natural Short Transaction object has failed due to a coding error.

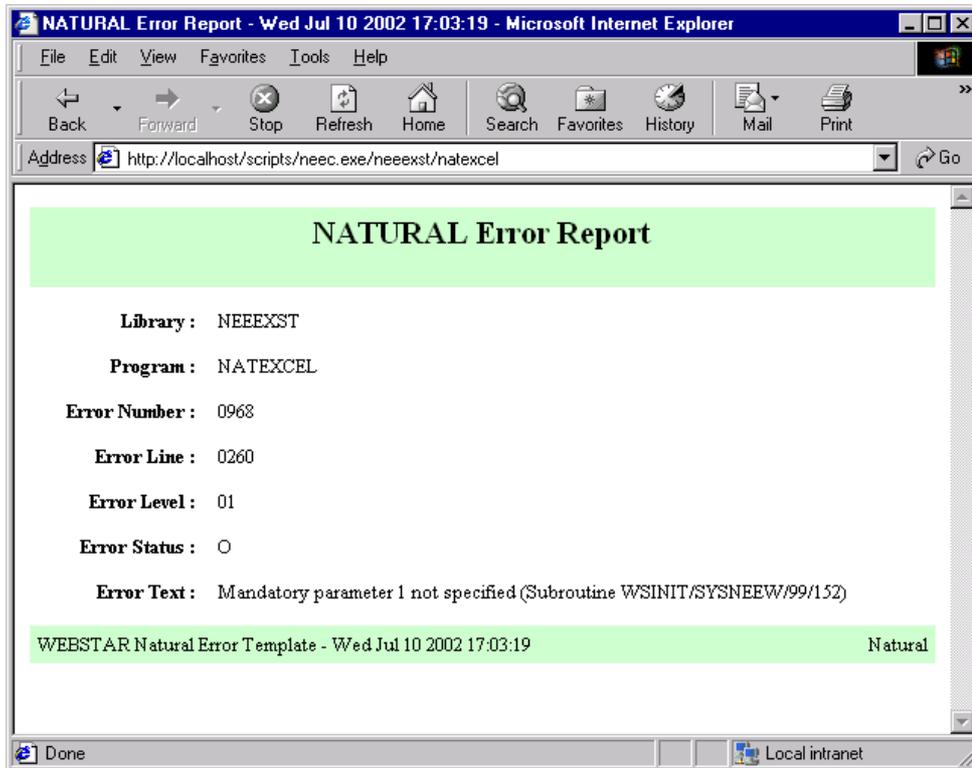


Figure 5-2 Error Report for Natural Short Transaction coding error

## Web Server Components

Web page templates need to be created to present the Short Transaction results data in a web browser. The web page templates can be created using HTML or XSL markup languages. Each template will contain any required page styles and data references used to merge the data from the Natural Short Transaction object with the web page template.

The choice of markup language, HTML or XSL largely depends on the web server and web browser technology to be used. If either the web server or web browser support XML/XSL data merging, then the XSL template will be utilized, otherwise, the HTML template will be utilized.

### Creating Page Templates

HTML or XSL page templates can be created using a text editor (or any suitable HTML/XSL editors). It is recommended that they be saved using a file name that reflects the Short Transaction object name. For example:

For a Short Transaction object NATEMP, a web page template could be saved as either 'natemp.html' or 'natemp.xml'.

Any templates created should be saved in a folder that is named the same as the Natural library from where the Short Transaction objects will execute. For example:

Short Transaction NATEMP resides on Natural library NEEEXST. The web page templates 'natemp.html' / 'natemp.xml' are stored in a folder named 'Neeexst' on the web server.

*Note: The template folders must reside on the web server under the NEE folder. For more information refer to Chapter 2 Installation and Chapter 3 Configuration.*

## Short Transaction Data Substitution

Any Short Transaction results that are to be displayed using web page templates are output using 'data name pairs'. The data name pairs consist of a data name and a data value. This method is known as Data Substitution.

The Data Substitution area within a web page template needs to be specified using name identities, which will be used to merge the data from the Natural Short Transaction object.

*Note: The name used in the template must be identical to the data name coded in the Natural Short Transaction object. For more information refer to the section [Output Short Transaction Data using Data Substitution](#).*

There are two possible types:

- Single Occurrence.
- Multiple Occurrence.

### Single Occurrence

Single occurrence Data Substitution caters for any non-repeating data, for example a record count or message at the end of processing.

The format of the name identities within the web page template will depend on the markup language being used.

#### HTML Web Pages

The data identity is coded delimited by '\$' at the start and end of the data identity. For example:

HTML code:

```
<p>Number of Records read = $COUNT$</p>
```

Natural code:

```
0450 PERFORM WS-WRITE-ENVIRONMENT #ETB 'COUNT' #COUNT
```

The Short Transaction will look for the data name 'COUNT' in the HTML page and replace it with the value in #COUNT.

# 5

## Natural Engineer WebStar

### XSL Web Pages

The data identity is coded using the `<xsl:value-of select="data identity" />` instruction. For example:

XSL code:

```
<p>Number of Records read = <xsl:value-of select="COUNT" /></p>
```

Natural code:

```
0450 PERFORM WS-WRITE-ENVIRONMENT #ETB 'COUNT' #COUNT
```

The Short Transaction will look for the data name 'COUNT' in the XSL page and replace it with the value in #COUNT.

### Multiple Occurrence

Multiple occurrence Data Substitution caters for repeating data, for example records from a database file or a list of objects from a Natural library.

For multiple occurrence Data Substitution, in addition to coding the name identities there is a second consideration required to provide a mechanism to loop round the Data Substitution area within the web page template until end of data is reached. This can be achieved by using NEE tags at the start and end of the Data Substitution area.

The format of the name identities and tags within the web page template will depend on the markup language being used.

### HTML Web Pages

The NEE tags need to be specified at the start and end of each Data Substitution area. These consist of two pairs:

Tags	Description
<code>&lt;NEE&gt;</code> <code>&lt;/NEE&gt;</code>	These are mandatory and are used by the WebStar run time process to identify the Data Substitution area. This pair of tags must only appear once within a single HTML page.
<code>&lt;NEE_LINE&gt;</code> <code>&lt;/NEE_LINE&gt;</code>	These are used for each block of repeating data within an HTML page. The names for these tags can be user defined, the values shown are examples only. These tag names must be unique within an HTML page, for example: NEE_LINE1, NEE_LINE2 would be required if using two repeating blocks.

The data identity is coded delimited by '\$' at the start and end of the data identity, within each repeating block used.

Example:

HTML code:

```
<NEE>
  <NEE_LINE>
    <tr>
      <td>$ID$</td>
      <td>$FIRST-NAME$</td>
      <td>$NAME$</td>
      <td>$BIRTH$</td>
    </tr>
  </NEE_LINE>
</NEE>
```

Natural code:

```
0300 READ (10) EMP-FILE
0310 /* INITIALIZE THE HTML NEE TAG
0320 PERFORM WS-WRITE-ENVIRONMENT #ETB 'HTML-NEE-TAG' 'NEE_LINE'
0330 /* OUTPUT THE DATA NAME & VALUE PAIRS
0340 PERFORM WS-WRITE-ENVIRONMENT #ETB 'ID' EMP-FILE.PERSONNEL-ID
0350 PERFORM WS-WRITE-ENVIRONMENT
0360 #ETB 'FIRST-NAME' EMP-FILE.FIRST-NAME
0370 PERFORM WS-WRITE-ENVIRONMENT #ETB 'NAME' EMP-FILE.NAME
0380 PERFORM WS-WRITE-ENVIRONMENT #ETB 'BIRTH' EMP-FILE.BIRTH
0390 ADD 1 TO #COUNT
0400 END-READ
```

Before the Natural Short Transaction object outputs any data, the first output must be to identify the correct template block where the multiple occurrences are to go. This is done at line **0320**, where a data name 'HTML-NEE-TAG' (for internal use) and data value '<NEE\_LINE>':

The actual data for the READ can now be output, this is done at lines **0340** to **0380**. Processing will take each data pair being output and match each value with the respective data name in the HTML page template.

### XSL Web Pages

The same specifications apply as for HTML web pages, with the following differences:

- The tags used to identify the Data Substitution area within the XSL page are defined using the `<xsl:for-each select="NEE">` and `</xsl:for each>` instructions.
- The tags used to identify each block of repeating data within the Data Substitution area are defined using the `<xsl:for-each select="NEE_LINE">` and `</xsl:for each>` instructions.

Example:

XSL code:

```
<xsl:for-each select="NEE">
<xsl:for-each select="NEE_LINE">
  <tr>
    <td><xsl:value-of select="ID" /></td>
    <td><xsl:value-of select="FIRST-NAME" /></td>
    <td><xsl:value-of select="NAME" /></td>
    <td><xsl:value-of select="BIRTH" /></td>
  </tr>
</xsl:for-each>
</xsl:for-each>
```

Natural code:

```
0300 READ (10) EMP-FILE
0310 /* INITIALIZE THE HTML NEE TAG
0320 PERFORM WS-WRITE-ENVIRONMENT #ETB 'HTML-NEE-TAG' 'NEE_LINE'
0330 /* OUTPUT THE DATA NAME & VALUE PAIRS
0340 PERFORM WS-WRITE-ENVIRONMENT #ETB 'ID' EMP-FILE.PERSONNEL-ID
0350 PERFORM WS-WRITE-ENVIRONMENT
0360 #ETB 'FIRST-NAME' EMP-FILE.FIRST-NAME
0370 PERFORM WS-WRITE-ENVIRONMENT #ETB 'NAME' EMP-FILE.NAME
0380 PERFORM WS-WRITE-ENVIRONMENT #ETB 'BIRTH' EMP-FILE.BIRTH
0390 ADD 1 TO #COUNT
0400 END-READ
```

Before the Natural Short Transaction object outputs any data, the first output must be to identify the correct template block where the multiple occurrences are to go. This is done at line **0320**, where a data name 'HTML-NEE-TAG' (for internal use) and data value '<NEE\_LINE>'.

The actual data for the READ can now be output, this is done at lines **0340** to **0380**. Processing will take each data pair being output and match each value with the respective data name in the XSL page template.

## Linking to other Short Transactions

It is possible to link from one set of Short Transaction results to another Short Transaction. An example of this would be running a Short Transaction to list objects within a Natural library and then listing the source code for an individual object by selecting the object from the list. When the object is selected, this will trigger a second Short Transaction, which will result in the source code list being produced.

*Note: This can be viewed by running the supplied example Short Transaction NATDIR. For more information refer to section [Supplied Example Short Transaction Objects](#).*

Linking to another Natural Short Transaction can be achieved by using the Anchor tag '<A>'.

HTML code based on the supplied example HTML page template 'natdir.html':

```
<tr bgcolor="$BGCOLOR$">
  <td><a href="natsrc?LIB=$HLIB$&OBJ=$HOBJ$" >$OBJECT$</a></td>
<td>$OBJECT-DESC$</td>
</tr>
```

The anchor tag contains a link to the Natural Short Transaction object NATSRC using the 'href=' attribute. This will pass the input data parameters required by NATSRC using the string '?LIB=\$HLIB\$&OBJ=\$HOBJ\$'. This parameter string will dynamically replace the data for '\$HLIB\$' with a library name and '\$HOBJ\$' with the selected object name.

XSL code based on the supplied example XSL page template 'natdir.xsl':

```
<tr bgcolor="{ $BGCOLOR } ">
  <td><a href="natsrc?LIB={ $HLIB } & OBJ={ $HOBJ } ">
    <xsl:value-of select="OBJECT" /></a>
  </td>
  <td><xsl:value-of select="OBJECT-DESC" /></td>
</tr>
```

The anchor tag contains a link to the Natural Short Transaction object NATSRC using the 'href=' attribute. This will pass the input data parameters required by NATSRC using the string '?LIB={ \$HLIB } & OBJ={ \$HOBJ }'. This parameter string will dynamically replace the data for '{ \$HLIB }' with a library name and '{ \$HOBJ }' with the selected object name.

## Supplied Example Web Page Templates

The Natural Engineer add-on component WebStar comes supplied with a set of HTML and XSL template pages. The supplied web pages can be modified or copied to customize their appearance.

*Note: The supplied templates complement the supplied example Short Transaction objects. For more information refer to section [Supplied Example Short Transaction Objects](#). The supplied templates are located in the NEEEXST folder under the web server NEE folder. For more information refer to Chapter 2 Installation and Chapter 3 Configuration.*

The following web page templates are supplied:

Object	Description
<b>Natdir.html</b>	HTML template used for producing a list of objects within a Natural library. Used by the supplied example Short Transaction object <a href="#">NATDIR</a> .
<b>Natdir.xsl</b>	XSL template used for producing a list of objects within a Natural library. Used by the supplied example Short Transaction object <a href="#">NATDIR</a> .
<b>Natemp.html</b>	HTML template used to produce a list of EMPLOYEES records. Used by the supplied example Short Transaction objects <a href="#">NATEMP</a> and <a href="#">NATEMPN</a> .
<b>Natemp.xsl</b>	XSL template used to produce a list of EMPLOYEES records. Used by the supplied example Short Transaction objects <a href="#">NATEMP</a> and <a href="#">NATEMPN</a> .
<b>Natsrc.html</b>	HTML template used for producing a source code listing for a specified or selected object. Used by the supplied example Short Transaction object <a href="#">NATSRC</a> .
<b>Natsrc.xsl</b>	XSL template used for producing a source code listing for a specified or selected object. Used by the supplied example Short Transaction object <a href="#">NATSRC</a> .

## Executing Short Transactions

---

This section describes how to execute Short Transactions via the Internet.

### EntireX Broker

The EntireX Broker Instance being used for the application must be invoked using the EntireX Broker Administration function and selecting the Broker to be used, for example ETB255.

### Attach Manager (NEEATM)

#### Windows

The Attach Manager must be invoked by executing object NEEATM.EXE. This is located in the X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\WEB\NEEATM folder (where X: is the directory where Natural Engineer has been installed). Typically this would be invoked using the START → RUN command line.

To start the Attach Manager:

```
"X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\WEB\NEEATM\NEEATM.EXE"
```

To terminate the Attach Manager:

```
"X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2\WEB\NEEATM\NEEATM.EXE  
END"
```

## 5

### Natural Engineer WebStar

#### CICS and Com-Plete

Log on to Natural and then log on to library NEEATM. The following commands can be used:

To start the Attach Manager:

```
NEEATM START
```

To terminate the Attach Manager:

```
NEEATM END
```

To determine the status of the Attach Manager:

```
NEEATM STATUS
```

#### Sun Solaris 8

*Note: The documentation relates to the Unix environment where case sensitivity is important. Any described input settings should be treated as read.*

The Attach Manager must be invoked by executing object NEEATM.EXE. This is located in the '/sag/nws/neeatm' directory. Typically this would be invoked using the Terminal Window command line.

To start the Attach Manager:

```
./neeatm.exe > neeatm.log 2>&1 &
```

To terminate the Attach Manager:

```
./neeatm.exe END
```

## Web Browser

### General

Open a web browser and type in the URL based on the following:

```
http://www.myweb.com/scripts/nec.exe/library-name/object-name
```

where:

'www.myweb.com' is the address of the web server, or a TCP/IP address.

'library-name' is the name of the Natural library containing the Short Transaction.

'object-name' is the name of the Short Transaction object to be executed.

This will now display the results for the Short Transaction in the web browser.

### SMARTS HTTP Server

If using the SMARTS HTTP Server environment, the Short Transactions can be invoked using two different methods:

#### 1. Using the WebStar web interface program NEEC.

This will invoke the WebStar web interface program to provide the interface between Natural and the web server.

```
http://www.myweb.com/scripts/nec.exe/library-name/object-name
```

where:

'www.myweb.com' is the address of the web server, or a TCP/IP address.

'library-name' is the name of the Natural library containing the Short Transaction.

'object-name' is the name of the Short Transaction object to be executed.

This will now display the results for the Short Transaction in the web browser.

# 5

## Natural Engineer WebStar

### 2. Using the Natural web interface program.

This will invoke the Natural CGI program to provide the interface between Natural and the web server.

Open a web browser and type in the URL based on the following:

```
http://www.myweb.com/natcgi/sysneew/neeaps/library-name/object-name
```

where:

'www.myweb.com' is the address of the web server, or a TCP/IP address.

'library-name' is the name of the Natural library containing the Short Transaction.

'object-name' is the name of the Short Transaction object to be executed.

This will now display the results for the Short Transaction in the web browser.

### Restrictions with SMARTS HTTP Server

Using the SMARTS HTTP server has the following restrictions:

1. Using the Natural Web Interface program will only support XSL template pages.

## Example Short Transaction

---

This example will illustrate the processes involved in utilizing the WebStar Short Transaction Processing.

A Natural program '**MYQUERY**' will be created, which will perform a simple read of 10 records from the EMPLOYEES database file and display the record details for PERSONNEL-ID, FIRST-NAME, NAME and BIRTH using a HTML template page.

The example is based on the Natural object NATEMP, which can be found in the supplied Short Transaction example library NEEEXST.

The web page templates referenced are '**natemp.html**' and '**natemp.xml**', which can be found in the supplied example NEEEXST folder under the WEB folder located in X:\PROGRAM FILES\SOFTWARE AG\NATURAL ENGINEER\4.4.2 folder (where X: is the directory on which Natural Engineer was installed).

*Note: This example is based on a Windows environment. The Natural environment being used must have the correct steplib structure defined in the Natural Parameter File. This example will be referencing objects that are located in the WebStar run time library SYSNEEW.*

*The example will assume that the Natural library NEEEXST is being used to create the example object. In order for any Short Transaction Natural objects to Stow and execute correctly the following steplib structure should be in place:*

*NEEEXST  
SYSNEEW*

**Step 1** Create a new program object called '**MYQUERY**' with the following comment section:

```
0010 /*****
0020 /* OBJECT      : MYQUERY                *
0030 /*
0040 /* FUNCTION : READ 10 RECORDS FROM THE EMPLOYEES FILE      *
0050 /*              AND DISPLAY DETAILS USING HTML/XSL TEMPLATE *
0060 /*****
```

# 5

## Natural Engineer WebStar

### Step 2 Data Definitions.

The first definition will be to include the data buffer area used to pass the Short Transaction results data between the Natural Server session and the web server via EntireX Broker. This is achieved by including NEEETB-A as the first entry.

Then the logical view of the EMPLOYEES file is defined specifying the fields PERSONNEL-ID, FIRST-NAME, NAME and BIRTH. This will be the data that will be displayed for each record read.

Finally a variable used to keep a count of the records read is defined.

```
0070 DEFINE DATA
0080 LOCAL USING NEEETB-A
0090 LOCAL
0100 01 EMP-FILE VIEW EMPLOYEES
0110   02 PERSONNEL-ID
0120   02 FIRST-NAME
0130   02 NAME
0140   02 BIRTH
0150 01 #COUNT (I02) INIT <+0>
0160 END-DEFINE
```

### Step 3 Initialize the data buffer area.

This is done by performing the external routine **WS-INIT** and passing the data buffer area **#ETB**.

```
0170 /* -----
0180 /* INITIALIZE ETB BUFFER
0190 /* -----
0200 PERFORM WS-INIT #ETB
```

### Step 4 Initialize the start of Short Transaction output.

The data area buffer needs to be initialized so that the correct web page template is used to format the output data. This example is going to use the templates '**natemp.html**' and '**natemp.xml**'.

This is done by performing the external routine **WS-WRITE-SHORT-INIT** and passing the data buffer area **#ETB**, the file name of the template '**natemp**' (the file extensions 'html' or 'xml' are not required) and '**I**' (to indicate the start of data output).

```
0210 /* -----
0220 /* INITIALIZE THE START OF SHORT TRANSACTION OUTPUT
0230 /* -----
0240 PERFORM WS-WRITE-SHORT-INIT #ETB 'NATEMP' 'I'
```

**Step 5** Read the EMPLOYEES database file and output the record details.

The EMPLOYEES database file is going to be read and 10 records retrieved. For each record read, the data for fields PERSONNEL-ID, FIRST-NAME, NAME and BIRTH are going to be output.

This data needs to be formatted as data name and value pairs, which will then be substituted within the HTML page template. This is done by performing the external routine **WS-WRITE-ENVIRONMENT** and passing the data buffer area **#ETB**, the data name and data value pairs.

```

0250 /* -----
0260 /* READ 10 EMPLOYEE RECORDS AND DISPLAY INFORMATION
0270 /* -----
0280 READ (10) EMP-FILE
0290 /* -----
0300 /* INITIALIZE THE HTML NEE TAG
0310 /* -----
0320 PERFORM WS-WRITE-ENVIRONMENT #ETB 'HTML-NEE-TAG' 'NEE_LINE'
0330 /* -----
0340 /* OUTPUT THE DATA NAME & VALUE PAIRS
0350 /* -----
0360 PERFORM WS-WRITE-ENVIRONMENT #ETB 'ID' EMP-FILE.PERSONNEL-ID
0370 PERFORM WS-WRITE-ENVIRONMENT #ETB 'FIRST-NAME'
0380 EMP-FILE.FIRST-NAME
0390 PERFORM WS-WRITE-ENVIRONMENT #ETB 'NAME' EMP-FILE.NAME
0400 PERFORM WS-WRITE-ENVIRONMENT #ETB 'BIRTH' EMP-FILE.BIRTH
0410 ADD 1 TO #COUNT
0420 END-READ

```

In order for the data name and value pairs to be correctly substituted in the web page, the HTML NEE tag needs to be identified. This is done using the statement at line 0320 A dummy data name of **'HTML-NEE-TAG'** and data value **'NEE\_LINE'** is used, which will locate the tag name **'<NEE\_LINE>'** within the template **'natemp'**. This will allow the subsequent output to correctly position the data name and value pairs.

**Step 6** Initialize the end of Short Transaction output.

All the data output has now been completed. This is done by performing the external routine **WS-WRITE-SHORT-INIT** and passing the data buffer area **#ETB**, the file name of the template **'natemp'** and **' '** (blank) to indicate the end of data output.

```

0430 /* -----
0440 /* INITIALIZE THE END OF SHORT TRANSACTION OUTPUT
0450 /* -----
0460 PERFORM WS-WRITE-SHORT-INIT #ETB 'NATEMP' ' '

```

# 5

## Natural Engineer WebStar

### Step 7 Output the record count

The record count is output by performing the external routine **WS-WRITE-ENVIRONMENT** and passing the data buffer area **#ETB**, the data name **'COUNT'** and data value **#COUNT**.

```
0470 /* -----  
0480 /* OUTPUT RECORD COUNT  
0490 /* -----  
0500 PERFORM WS-WRITE-ENVIRONMENT #ETB 'COUNT' #COUNT
```

### Step 8 Close the data buffer area.

This is done by performing the external routine **WS-END** and passing the data buffer area **#ETB**.

```
0510 /* -----  
0520 /* CLOSE ETB BUFFER  
0530 /* -----  
0540 PERFORM WS-END #ETB  
0550 /*  
0560 END
```

This completes all the code required for this Natural Short Transaction object. The complete source code looks like this:

```
0010 /*****  
0020 /* OBJECT      : MYQUERY *  
0030 /* *  
0040 /* FUNCTION : READ 10 RECORDS FROM THE EMPLOYEES FILE *  
0050 /*          AND DISPLAY DETAILS USING HTML/XSL TEMPLATE *  
0060 /*****  
0070 DEFINE DATA  
0080 LOCAL USING NEEETB-A  
0090 LOCAL  
0100 01 EMP-FILE VIEW EMPLOYEES  
0110 02 PERSONNEL-ID  
0120 02 FIRST-NAME  
0130 02 NAME  
0140 02 BIRTH  
0150 01 #COUNT (I02) INIT <+0>  
0160 END-DEFINE
```

```
0170 /* -----
0180 /* INITIALIZE ETB BUFFER
0190 /* -----
0200 PERFORM WS-INIT #ETB
0210 /* -----
0220 /* INITIALIZE THE START OF SHORT TRANSACTION OUTPUT
0230 /* -----
0240 PERFORM WS-WRITE-SHORT-INIT #ETB 'NATEMP' 'I'
0250 /* -----
0260 /* READ 10 EMPLOYEE RECORDS AND DISPLAY INFORMATION
0270 /* -----
0280 READ (10) EMP-FILE
0290 /* -----
0300 /* INITIALIZE THE HTML NEE TAG
0310 /* -----
0320 PERFORM WS-WRITE-ENVIRONMENT #ETB 'HTML-NEE-TAG' 'NEE_LINE'
0330 /* -----
0340 /* OUTPUT THE DATA NAME & VALUE PAIRS
0350 /* -----
0360 PERFORM WS-WRITE-ENVIRONMENT #ETB 'ID' EMP-FILE.PERSONNEL-ID
0370 PERFORM WS-WRITE-ENVIRONMENT #ETB 'FIRST-NAME'
0380 EMP-FILE.FIRST-NAME
0390 PERFORM WS-WRITE-ENVIRONMENT #ETB 'NAME' EMP-FILE.NAME
0400 PERFORM WS-WRITE-ENVIRONMENT #ETB 'BIRTH' EMP-FILE.BIRTH
0410 ADD 1 TO #COUNT
0420 END-READ
0430 /* -----
0440 /* INITIALIZE THE END OF SHORT TRANSACTION OUTPUT
0450 /* -----
0460 PERFORM WS-WRITE-SHORT-INIT #ETB 'NATEMP' ' '
0470 /* -----
0480 /* OUTPUT RECORD COUNT
0490 /* -----
0500 PERFORM WS-WRITE-ENVIRONMENT #ETB 'COUNT' #COUNT
0510 /* -----
0520 /* CLOSE ETB BUFFER
0530 /* -----
0540 PERFORM WS-END #ETB
0550 /*
0560 END
```

The object **MYQUERY** can now be stowed and is ready for execution.

**Step 9** The web page templates.

There are two web page templates available for this example: '**natemp.html**' (HTML format) and '**natemp.xml**' (XSL format). Which one is referenced will be determined at run time. If the web server or web browser supports XML/XSL data merging then the template '**natemp.xml**' will be used, otherwise the template '**natemp.html**' will be used.

HTML page template '**natemp.html**'.

The data name and value pairs, bound by the HTML NEE tag **<NEE\_LINE>** are marked in bold. This is the area where the record data will be output. All the data names are delimited by '\$', these will be substituted by the data values passed from the Natural Short Transaction object. The record count will be substituted by the data identity '**\$COUNTS**'.

```
<html>
<head>
  <title>READ ADABAS DATA</title>
</head>

<body>
  <h2>Display EMPLOYEES Details</h2>

  <table width='100%' border='0' cellpadding='2' cellspacing='2'>

    <tr bgcolor='#99CCFF'>
      <th>ID</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Date of Birth</th>
    </tr>

    <NEE>
    <b>NEE_LINE</b>
      <tr>
        <td>$ID$</td>
        <td>$FIRST-NAME$</td>
        <td>$NAME$</td>
        <td>$BIRTH$</td>
      </tr>
    </NEE_LINE>
    </NEE>

  </table>
  <HR></HR>
  <p>Number of Records read = $COUNT$</p>
</body>
</html>
```

XSL page template 'natemp.xsl'.

The data name and value pairs, bound by the HTML NEE tag **<xsl:for-each select="NEE\_LINE">** are marked in bold. This is the area where the record data will be output. All the data names are identified by the **<xsl:value-of >** instructions, these will be substituted by the data values passed from the Natural Short Transaction object. The record count will be substituted by the data identity **"COUNT"**.

```
<?xml version='1.0' encoding="iso-8859-1" ?>
<xsl:stylesheet version='1.0'
xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
<xsl:output method="html" indent="yes" />

<xsl:template match="OUTPUT">
<html>
<head>
  <title>READ ADABAS DATA</title>
</head>
<body>

  <h2><font color="red">Display EMPLOYEES Details</font></h2>

  <table width='100%' border='0' cellpadding='5' cellspacing='3'>
    <tr bgcolor="#99CCFF">
      <th>Personnel ID</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Date of Birth</th>
    </tr>

    <xsl:for-each select="NEE">
      <xsl:for-each select="NEE_LINE">
        <tr>
          <td><xsl:value-of select="ID" /></td>
          <td><xsl:value-of select="FIRST-NAME" /></td>
          <td><xsl:value-of select="NAME" /></td>
          <td><xsl:value-of select="BIRTH" /></td>
        </tr>
      </xsl:for-each>
    </xsl:for-each>
  </table>

  <p>Number of Records read = <xsl:value-of select="COUNT" /> />
</p>

</body>
</html>
```

# 5

## Natural Engineer WebStar

**Step 10** Executing the Short Transaction over the Internet.

1. Start the Broker session. For this example Broker ETB255 is being used.

The following Figure 5-3 illustrates the EntireX Broker ETB255 started.

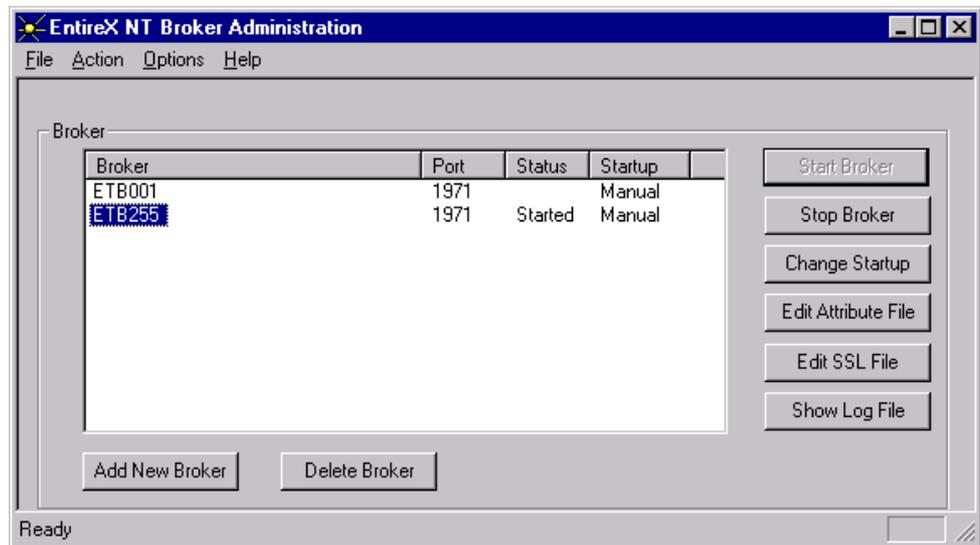
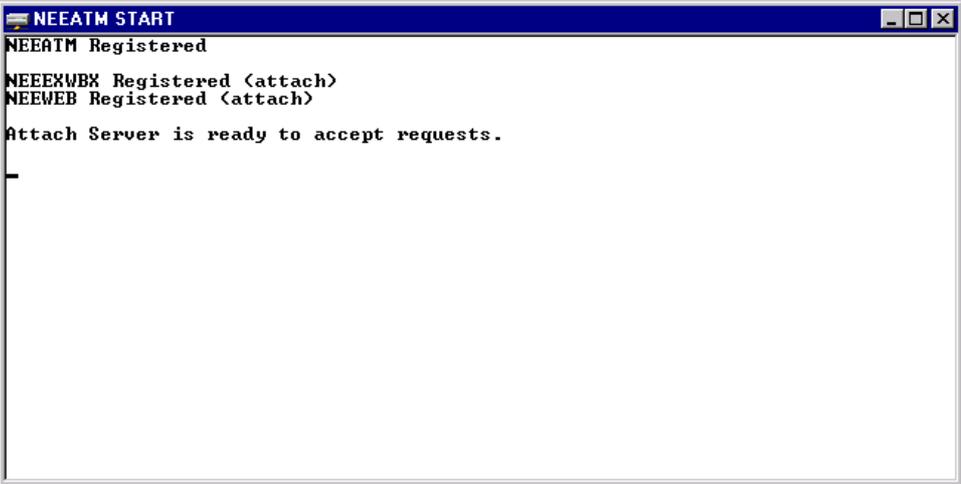


Figure 5-3 EntireX Broker ETB255 started

2. Start the Attach Manager NEEATM.

The following Figure 5-4 illustrates the Attach Manager NEEATM start screen.



```
NEEATM START
NEEATM Registered
NEEEXWBX Registered (attach)
NEEWEB Registered (attach)
Attach Server is ready to accept requests.
```

Figure 5-4 Attach Manager NEEATM start screen

# 5

## Natural Engineer WebStar

3. Open the web browser and type in the Short Transaction URL:

**`http://localhost/scripts/nec.exe/necxst/myquery.`**

*Note: 'localhost' will need to be the correct IP address of the web server you are using.*

The following Figure 5-5 illustrates the web browser with the Short Transaction URL.

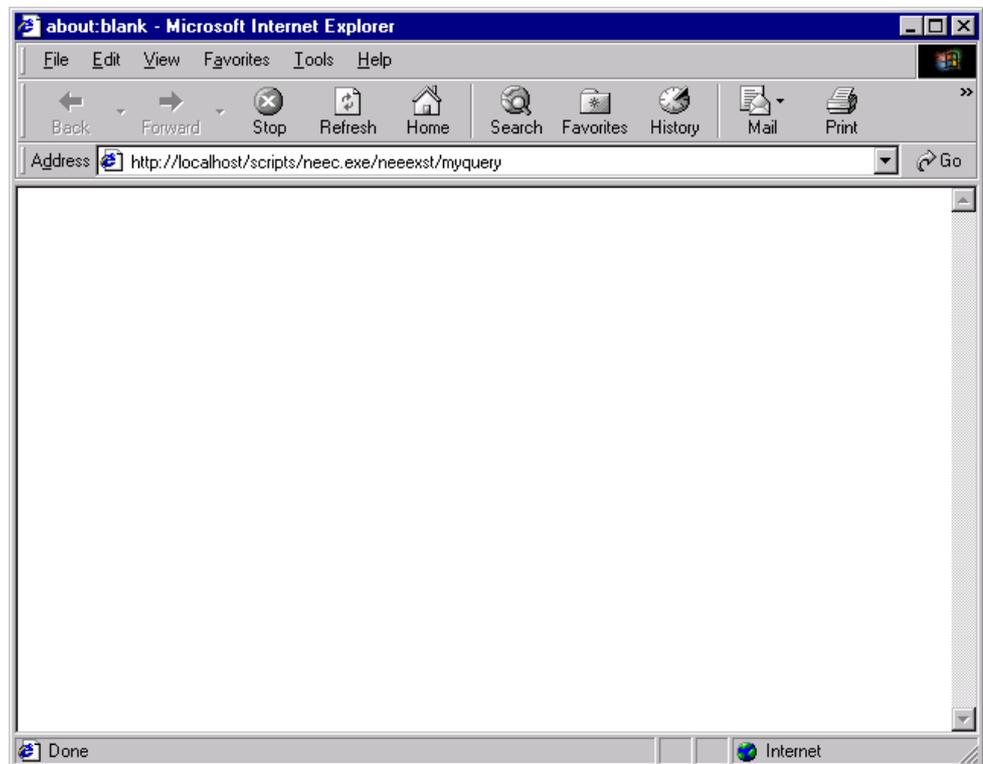
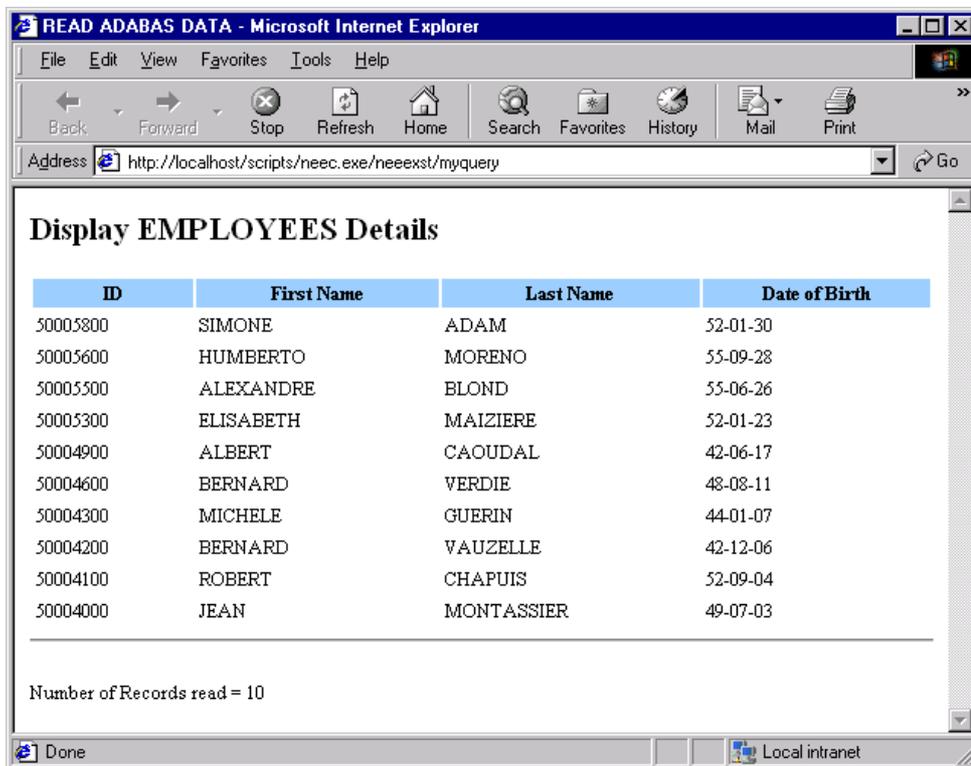


Figure 5-5 Web Browser with the Short Transaction URL

4. The web page template '**natemp.xml**' (if the web server or web browser supports XML/XSL) otherwise the web page template '**natemp.html**' will now be displayed in the web browser showing the record details.

The following Figure 5-6 illustrates the results for the Natural Short Transaction '**MYQUERY**'.



ID	First Name	Last Name	Date of Birth
50005800	SIMONE	ADAM	52-01-30
50005600	HUMBERTO	MORENO	55-09-28
50005500	ALEXANDRE	BLOND	55-06-26
50005300	ELISABETH	MAIZIERE	52-01-23
50004900	ALBERT	CAUDAL	42-06-17
50004600	BERNARD	VERDIE	48-08-11
50004300	MICHELE	GUERIN	44-01-07
50004200	BERNARD	VAUZELLE	42-12-06
50004100	ROBERT	CHAPUIS	52-09-04
50004000	JEAN	MONTASSIER	49-07-03

Number of Records read = 10

Figure 5-6 The results for the Natural Short Transaction '**MYQUERY**'



# INDEX

## A

- Applying the WebStar Development Function under OS/390, 104
  - Environment Administration, 107
  - Extract the Application, 105
  - Impact Analysis, 106
  - Load Repository, 105
  - Modification, 108
  - Process Overview, 104
- Applying the WebStar Development Function under Windows, 98
  - Extract the Application, 99
  - Impact Analysis, 100
  - Load Repository, 99
  - Modification, 103
  - Process Overview, 98
  - WebStar Modification Preferences, 101

## C

- Configuration, 55
  - Configuration Options, 56
    - Infrastructure, 56
    - Product and Platform Support, 56
- EntireX Communicator Components, 80
  - Broker, 80
  - Broker Services, 83
  - Broker Stub, 84
- Natural, 85
  - Natural Parameter Files, 92
  - Natural Security, 96
  - NEEATM, 85
  - NEEINI-S, 90
  - User Exit Considerations, 96
  - WebStar Initialization Settings, 95

- Web Server, 58
  - Broker Stub, 69
  - Configuration Settings, 70
  - HTML/XSL Folders, 58
  - NEEATM, 66
  - NEEC.EXE, 59

## E

- Executing Short Transactions, 175
  - Attach Manager, 175
  - EntireX Broker, 175
  - Web Browser, 177
- Executing the Web Enabled Natural Application, 113
  - Attach Manager, 113
  - EntireX Broker, 113
  - Web Browser, 115

## I

- Installation, 37
  - OS/390 and VSE/ESA, 40
    - Copying the Tape Contents to Disk for OS/390, 42
    - Copying the Tape Contents to Disk for VSE/ESA, 44
    - Installation Steps for OS/390, 43
    - Installation Steps for VSE/ESA, 46
    - Prerequisites, 40
    - Tape Contents for OS/390, 41
    - Tape Contents for VSE/ESA, 44
    - Using System Maintenance Aid, 40
  - SMARTS under OS/390, 50
    - Copying the Tape Contents to Disk for SMARTS under OS/390, 52

## **Natural Engineer WebStar**

- Installation Steps for SMARTS under OS/390, 53
- Prerequisites, 50
- Tape Contents for SMARTS under OS/390, 51
- Using System Maintenance Aid, 50
- Solaris 8, 47
  - Installation Process, 47
  - Prerequisites, 47
- Windows, 38
  - Installation Process, 39
  - Prerequisites, 38

## **M**

- Modifying Application Objects, 19
  - Generated Web Pages, 24
  - Modified Natural Objects, 19
  - Multiple View Data, 28
  - Natural Engineer Process, 19
  - Single View Data, 25

## **O**

- Operating Environment, 11
  - Browser Communication, 12
  - Interface Software, 14
  - Overview, 18
  - Transaction Processing, 15
  - Web Pages, 17
  - Web Server Communication to Natural, 13

## **R**

- Reporting Mode Considerations, 120
  - Mode Conversion, 120
  - Natural Engineer Process Overview, 120

## **S**

- Short Transaction Processing, 143
  - Creating Short Transactions, 145
    - Change Content Type, 156
    - Coding Short Transactions, 147
    - Data Buffer Area, 148
    - Data Substitution, 157
    - End Short Transaction, 161
    - Free Format, 159
    - Initialize Data Buffer Area, 149
    - Initialize Short Transaction Output, 153
    - Overview, 145
    - Read Data Buffer Area, 152
    - Read Input Data Parameter Strings, 150
  - Error Processing, 166
  - Supplied Example Short Transactions, 162
  - Web Server Components, 168
    - Creating Page Templates, 168
    - Data Substitution, 169
    - Linking Short Transactions, 173
    - Supplied Example Templates, 174

## **W**

- Web Execution Process, 31
  - Overview, 32
  - Web Example, 33