

Test Tool

[WIN:

This Test Utility Web Online is a component of Natural Web Interface. You have the ability to check your subprogram locally without involving an HTTP server. The transfer parameters for your Web page are transferred into the Test Utility and are posted directly to the business logic. As communication platform, you can choose either RPC or DCOM as in real remote communications. The result is either the Web page expected or an error message. The Web page can be viewed with the browser or a viewer of your choice. If you receive an error message, you can easily debug your business logic locally without writing an extra test routine. No remote debugging is needed.

Features:

- Local application checking.
- No need for remote debugging.
- Simplified error checking.
- Comfortable operation by user friendly interface.
- No need to write an extra test routine.

Below is information on:

- Prerequisites
- Running the Application
- Supported Content-Types
- Menu
- Input/Output Fields
- Buttons

Prerequisites

- Web browser which supports different content types, for example, Microsoft Internet Explorer Version 5.0 or higher.
- Any available text editor.

Running the Application

To define path adjustments

1. Start the main dialog.
2. Select a browser and viewer of your choice with Tools > Options...
3. Set the browser, viewer and work file path.
4. Press the OK button.

To start the application

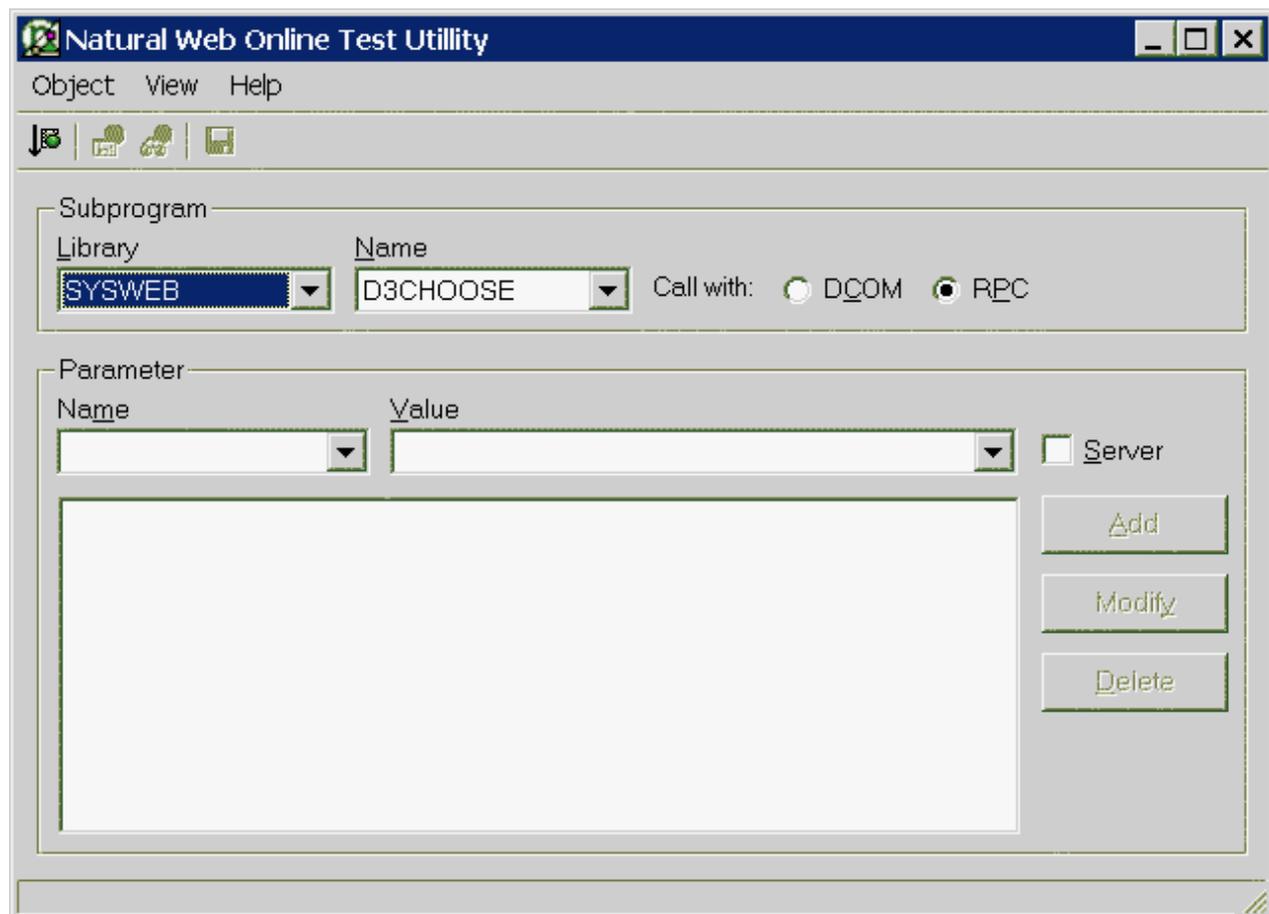
1. Start the dialog WEB-ONL.
2. Select a library and subprogram name.
3. Optional: add parameters.
4. Choose RPC or DCOM.
5. Press the Execute button.
6. View the result by pressing either the Result... or the Browse... button.

Supported Content Types

The following Content Types are supported by the Test Utility:

Content Type	Extension
"application/rtf"	"rtf"
"application/powerpoint"	"ppt"
"application/msword"	"doc"
"application/excel"	"xls"
"text/html"	"htm"
"text/plain"	"txt"
"text/xml"	"xml"
"text/richtext"	"rtf"

If you need further Content Types, change the subroutine HTML2CONTENT-TYPE (SYSWEB/W3CO2EXT) and extend the translation table to your own needs.



Menu

Object

Execute Subprogram

Starts the editor chosen with the Options dialog.

Save to Natural Text

Saves the returned data as Natural object of the type Text.

Exit

Leaves the dialog.

View

Result...

Executes the selected subprogram.

Browser...

Starts the browser chosen with the Options dialog.

Options...

Opens the Options dialog.

Help

Contents

Displays this HTML-based help file.

About

Provides general program information.

Input/Output Fields

Field	Explanation
Subprogram: Library Name	Enables you to specify the library and the name of the required subprogram. The available libraries and subprograms are automatically taken from the library workspace and listed in selection boxes.
Server	If any of the name-value-pairs are server variables, you need to check this toggle button. Note that any status will last until you change it again.
Call with	Can be selected with either DCOM or RPC as communication form. For DCOM, you have to register your classes first. Default: RPC
Parameter: Name Value	Here you can enter the name-value-pairs needed from the subprogram. To take them over into the parameter list, press the Add button. To modify the entries, use the Modify button. You do not have to substitute &, =, %; this will be done by the WEB-ONL program. If you use server parameters, check the Server toggle button before you add the parameter to the parameter list. In the parameter list, all name-value-pairs are displayed. &, =, % are substituted. To delete a pair, select the item and press the Delete button. Every selected item will be inserted into the Name and Value fields. If you wish to modify a pair, select the item, change it in the Name and Value fields and press the Modify button.

Buttons



Runs the process of receiving the output from the requested subprogram. The status of the process can be seen in the status bar at the bottom of the WEB-ONL dialog window.



Starts the editor. It is disabled as long as you have not executed the program and if you have not changed the subprogram library or name. You can choose the editor with the Options dialog.



Starts the browser chosen with the Options dialog. It is disabled as long as you have not executed the program and if you have not changed the subprogram library or name.



Call Natural Web Interface Online Test Utility (WEB-ONL) to run the generated program and show the generated result.

:WIN][UX:**Not applicable to mainframe systems.**

This section describes the use of HTML to Natural, a program that enables you to convert an HTML page into a Natural subprogram for use with Natural Web Interface.

Using HTML to Natural to generate Natural code from an HTML page avoids you having to adapt HTML input to the conventions of Natural code. You can then move the "HTML-page-turned-subprogram" to the server, including all the other Natural program logic you have added. If you want to change the HTML page again, go back to your source, convert it and move it to the server again. This is much easier than writing HTML with a browser, moving it to the server, adding Natural program logic and reiterating the process if your HTML page changes.

This section covers the following topics:

- Using the Conversion Program
- Inserting a Natural Tag
- Options
- Online Test Utility WEB-ONL

Using the Conversion Program

If your basic Web pages are designed with editing tools, it takes some effort to include such a page in a Natural subprogram that can be called from the Web.

"HTML to Natural" is a program that uses an HTML page as input and generates a Natural subprogram, which can be called by the Natural Web Server Extensions using the Natural Web Interface.

```

09:57:28          ***** HTML to Natural *****          2003-02-14
                   - Main Menu -                          Library SYSWEB

Input File:
/nat/natc/511/samples/sysweb/*.htm_____

Output to Natural
Library ..... SYSWEB
Object type ..... N
Object ..... _____
Subroutine name .. _____

Select HTML file for generation.
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           Exit           Class           Test  Opt.  Save  Canc

```

Below is information on:

- Functions and PF Keys
- Generating a Subprogram/Subroutine to be called direct from the Web

Functions and PF Keys

PF Key	Function	Explanation
PF1	Help	Invokes the Help function for the field at which the cursor is positioned.
PF3	Exit	Leaves the program and returns to the command line.
PF6	Class	Starts the program that generates a DCOM Class.
PF9	Test	Starts the Online Test Utility (see the relevant section).
PF10	Opt.	Options. Specifies options for the generation process.
PF11	Save	Saves the selected input and output files as default Natural parameters.
	Next	Starts generating the program.
ENTER		

Generating a Subprogram/Subroutine to be called direct from the Web

 **To generate a subprogram/subroutine to be called direct from the Web**

1. Select your HTML page.
2. Close the Natural library you want to generate.
3. Select the object type you want to generate.
4. Select your Natural file name.
5. Start the generation.
6. After generation, you can call the Natural Web Interface to show the page.

Inserting a Natural Tag

If you use Natural on your HTML page, it is possible to specify your special Natural coding direct in the HTML page. After generation, the program needs no additional changes.

The HTML2NAT program can recognize a <NATURAL> tag. All lines between <NATURAL> and </NATURAL> will be copied, as they are, to the generated Natural source object.

Appearance

```
<NATURAL> </NATURAL>
```

Below is information on:

- Attributes DATA, LDA, GDA, SUB, NOT
- Comment Tag
- ASP-like Script Commands
- Additional Script Directives
- Example 1 of a Simple Generation
- Example 2 of a Simple Generation with a Natural Tag

Attributes DATA, LDA, GDA, SUB, NOT

Listed below are attributes provided to define coding sections that are to be moved within the program or excluded from the program.

Attribute	Explanation
DATA	<NATURAL DATA> or <NATURAL LDA> moves the defined section to the DEFINE DATA LOCAL part of your program.
LDA	
GDA	<NATURAL GDA> moves the defined section to the DEFINE DATA GLOBAL part of your program.
SUB	<NATURAL SUB> moves the defined section to the end of the program. This enables you to specify inline subroutines.
NOT	<NATURAL NOT> excludes the defined section from the program. This enables you to specify the design of part of a page that will be generated by a program.

Comment Tag

Use the comment tag "<!-- -->" to hide the display of defined sections of your coding. If you use the comment tag and <NATURAL NOT>, you can display the predefined page with a normal browser. This helps you to specify your page and replace parts of the page dynamically.

ASP-like Script Commands

With the new Natural Version 5.1 of HTML2NAT, not only <NATURAL> and </NATURAL> can be used but also ASP-like (Active Server Page) script commands which are differentiated from the text by using the "<%" and "%>" delimiters.

Additional Script Directives

The following Natural-specific directives must be used when writing a Natural subprogram:

Output directive: <%= ... %>

Short form for <% PERFORM W3HTML ... %> tag

Subprogram directive: <%SUB ... %>

equal to the <NATURAL SUB> ... </NATURAL> tag

Global Data Area directive: <%GDA ... %>

equal to the <NATURAL GDA> ... </NATURAL> tag

directive: <%LDA ... %>

equal to the <NATURAL LDA> ... </NATURAL> tag

Not directive: <%NOT ... %>

equal to the <NATURAL NOT> ... </NATURAL> tag

Processing directive <%@ LANGUAGE=NATURAL %>

indicates that the used language is Natural..

Example 1 of a Simple Generation

HTML document:

```
<HTML><HEAD><TITLE>
Example1 genNat
</TITLE></HEAD><BODY><H2>
Example1 genNat
</H2><HR>
<P>This is for your output
</BODY></HTML>
```

Generated Natural subprogram:

```
* ----- SUBPROGRAM generated out of file:
* ----- C:\example1.html
DEFINE DATA
PARAMETER USING W3PARM
LOCAL USING W3CONST
LOCAL
* ----- PRIVATE VARIABLES -----
1 W3VALUE (A250)
END-DEFINE
*
* ----- ERROR HANDLER -----
ON ERROR
  PERFORM W3ERROR ##W3ERROR
  PERFORM W3END ##RPC
  ESCAPE ROUTINE
END-ERROR
* ----- INITIALIZE HTTP API -----
PERFORM W3INIT ##RPC
* ----- HEADER FOR SERVER -----
PERFORM W3CONTENT-TYPE 'text/html'
*
* ----- MAIN PROGRAM -----
PERFORM W3TEXTLINE '<HTML><HEAD><TITLE>'
PERFORM W3TEXTLINE 'Example genNat'
PERFORM W3TEXTLINE '</TITLE></HEAD><BODY><H2>'
PERFORM W3TEXTLINE 'Example genNat'
PERFORM W3TEXTLINE '</H2><HR>'
PERFORM W3TEXTLINE '<P>This is for your output'
PERFORM W3TEXTLINE '</BODY></HTML>'
* ----- END HTTP API -----
PERFORM W3END ##RPC
* ----- END MAIN PROGRAM -----
*
*
* ----- SUBROUTINES -----
END
```

Example 2 of a Simple Generation with a Natural Tag

HTML document:

```
<HTML><HEAD><TITLE>
Example2 genNat
</TITLE></HEAD><BODY><H2>
Example2 genNat
</H2><HR>
<P>This is for your output
<HR>
<P>generated at:
<NATURAL NOT>
Time/Date
</NATURAL>
<NATURAL><!--
  PERFORM DOTIME
--></NATURAL>
<NATURAL SUB><!--
DEFINE SUBROUTINE DOTIME
  COMPRESS *TIME *DATE INTO #VALUE
  PERFORM W3TEXTLINE #VALUE
END-SUBROUTINE
--></NATURAL>
<NATURAL DATA><!--
1 #VALUE (A30)
--></NATURAL>
</BODY></HTML>
```

Generated Natural subprogram:

```

* ----- SUBPROGRAM generated out of file:
* ----- C:\example2.html
DEFINE DATA
PARAMETER USING W3PARAM
LOCAL USING W3CONST
1 #VALUE (A30)
* ----- PRIVATE VARIABLES -----
1 W3VALUE (A250)
END-DEFINE
*
* ----- ERROR HANDLER -----
ON ERROR
  PERFORM W3ERROR ##W3ERROR
  PERFORM W3END ##RPC
  ESCAPE ROUTINE
END-ERROR
* ----- INITIALIZE HTTP API -----
PERFORM W3INIT ##RPC
* ----- HEADER FOR SERVER -----
PERFORM W3CONTENT-TYPE 'text/html'
*
* ----- MAIN PROGRAM -----
PERFORM W3TEXTLINE '<HTML><HEAD><TITLE>'
PERFORM W3TEXTLINE 'Example2 genNat'
PERFORM W3TEXTLINE '</TITLE></HEAD><BODY><H2>'
PERFORM W3TEXTLINE 'Example2 genNat'
PERFORM W3TEXTLINE '</H2><HR>'
PERFORM W3TEXTLINE '<P>This is for your output'
PERFORM W3TEXTLINE '<HR>'
PERFORM W3TEXTLINE '<P>generated at:'
  PERFORM DOTIME
PERFORM W3TEXTLINE '</BODY></HTML>'
* ----- END HTTP API -----
PERFORM W3END ##RPC
* ----- END MAIN PROGRAM -----
*
*
* ----- SUBROUTINES -----
DEFINE SUBROUTINE DOTIME
  COMPRESS *TIME *DATE INTO #VALUE
  PERFORM W3TEXTLINE #VALUE
END-SUBROUTINE
END

```

Note:

The syntax of the Natural program will not be checked during conversion.

Options

```
14:04:47          ***** HTML to Natural *****          2003-02-14
User SAG              - Options -                          Library SYSWEB

HTML File
Delete unnecessary white space ..... _
Save .. at source ..... _

Generated Source
Stow after generation ..... _
Natural line length ..... 128

Default input file:
$NATDIR/$NATVER/SAMPLES/SYSWEB/*.HTM_____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           Exit                                     Canc
```

Below is information on:

- Input/Output Fields
- Functions and PF Keys

Input/Output Fields

Field	Explanation
Delete unnecessary white space	If checked, multiple white-space characters such as blank, new line, tab, will be reduced to a single white space. For special HTML tags such as <PRE> <TEXTAREA> or <SCRIPT>, the white space will not be collapsed. Default value: unchecked
Save <NATURAL NOT> ... <NATURAL> in Source	If checked, the content of <NATURAL NOT> tags will not usually be generated into the Natural source. This option generates the content of <NATURAL NOT> as comment into the Natural source. Default value: unchecked
Stow after Generation	If checked, the generated program will be stowed if the generation has been successful. Default value: checked
Natural Line Length	The length of the generated Natural source lines: the minimum value is 20, the maximum 248. Default value: 72
Default Input File	The default input file to be used for the generation. Default value: /nat/natc/511/samples/sysext

Functions and PF Keys

PF Key	Function	Explanation
ENTER		Leaves the program and saves the changes.
PF3	Exit	Returns to the command line.
PF12	Canc	Leaves the program without saving your changes.

Online Test Utility WEB-ONL

This Test Utility is a component of Natural Web Interface. You have the ability to check your subprogram locally without involving an HTTP server. The transfer parameters for your Web page are transferred into the Test Utility and are posted directly to the business logic. As communication platform, you can choose either RPC or DCOM as in real remote communications. The result is either the Web page expected or an error message. The Web page can be viewed with the browser or a viewer of your choice. If you receive an error message, you can easily debug your business logic locally without writing an extra test routine. No remote debugging is needed.

Features:

- Local application check.
- No need for remote debugging.
- Simplified error checking.
- No need to write an extra test routine.

Below is information on:

- Running the Application
- Input/Output Fields
- Functions and PF Keys

Running the Application

▶ To run the application

1. Start the program WEB-ONL.
2. Select a library and subprogram name.
3. Optional: add parameters.
4. Choose RPC or DCOM.
5. Press ENTER.

```

09:55:24          ***** Natural Web Online Test Utility *****          2003-02-14
                                - Main Menu -                                Library SYSWEB

Library ..... SYSWEB__
Subprogram ..... NAT-ENV
DCOM/RPC ..... R
Output text object .. WEB-OUT_

Parameter
  Name                      Value
1: _____
2: _____
3: _____
4: _____
5: _____
6: _____
7: _____
8: _____
9: _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           Exit                                           Canc
    
```

Input/Output Fields

Field	Explanation
Library	The library in which the required subprogram is stored.
Subprogram	The name of the required subprogram.
DCOM/RPC	Can be selected with either DCOM or RPC as communication form. For DCOM, you have to register your classes first. Default: R
Output Text Object	The name of the Natural object of the type Text that stores the result of the generated Web page. Default: WEB-OUT
Parameter: Name Value	Here you can enter the name-value-pairs needed from the subprogram. If you use server parameters, place an ampersand (&) in front of the variable name before you add the parameter to the parameter list.

Functions and PF Keys

PF Key	Function	Explanation
ENTER		Runs the process of receiving the output of the requested subprogram. The status of the process can be seen in the message line at the bottom of the WEB-ONL program screen.
PF3	Exit	Leaves the Test Utility and returns to the command line.
PF12	Canc	Cancel. Stops processing.

:UX]