

Natural

Natural Web I/O Interface

Version 8.3.8

January 2017

This document applies to Natural Version 8.3.8.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 1992-2017 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: NATWIN-NNATWEBIO-838-20170102

Table of Contents

Preface	v
I Introduction	1
1 Introduction	3
What is the Natural Web I/O Interface?	4
Components of the Natural Web I/O Interface	4
Executing a Natural Application in a Web Browser	5
Client-Server Compatibility	6
Terminology	6
Differences in a SPoD Development Environment	7
Restrictions When Using the Natural Web I/O Interface with Natural Applications	7
Differences between the Natural Web I/O Interface Client and Terminal Emulation	9
II Installing and Configuring the Natural Web I/O Interface Server	11
2 Installing and Configuring the Natural Web I/O Interface Server	13
Installing the Natural Web I/O Interface Service	14
Configuring the Natural Web I/O Interface Service on Windows	14
Logging Information	19
SSL Support	20
III Installing the Natural Web I/O Interface Client	23
3 Prerequisites	25
Servlet Container	26
Natural for Mainframes	26
Natural for UNIX	26
Natural for OpenVMS	27
Natural for Windows	27
Browser Prerequisites	27
4 Installing the Natural Web I/O Interface Client on Apache Tomcat	29
Installation Steps	30
Installation Verification	32
5 Migrating the Natural Web I/O Interface Client from IIS to Apache Tomcat	33
Before You Install the Natural Web I/O Interface Client	34
Installing the Natural Web I/O Interface Client on Apache Tomcat	35
Configuring the Natural Web I/O Interface Client on Apache Tomcat	35
IV Configuring the Client	39
6 About the Logon Page	41
Starting a Natural Application from the Logon Page	42
Examples of Logon Pages	42
Dynamically Changing the CICS Transaction Name when Starting a Session	43
Specifying a Password in the Logon Page	44
Changing the Password in the Logon Page	44
Browser Restrictions	45

7	Natural Client Session Configuration	47
	General Information	48
	Name and Location of the Configuration File	48
8	Natural Client Configuration Tool	49
	Invoking the Configuration Tool	50
	Session Configuration	51
	Logging Configuration	59
	Logon Page	59
	Logout	60
9	Natural Web I/O Style Sheets	61
	Name and Location of the Style Sheets	62
	Editing the Style Sheets	62
	Modifying the Position of the Main Output and of the PF Keys	62
	Modifying the Font Size	64
	Modifying the Font Type	65
	Defining Underlined and Blinking Text	65
	Defining Italic Text	66
	Defining Bold Text	66
	Defining Different Styles for Output Fields	67
	Modifying the Natural Windows	67
	Modifying the Message Line	68
	Modifying the Background Color	68
	Modifying the Color Attributes	69
	Modifying the Style of the PF Key Buttons	70
	XSLT Files	70
10	Starting a Natural Application with a URL	73
11	Configuring Container-Managed Security	75
	General Information	76
	Name and Location of the Configuration File	76
	Activating Security	76
	Defining Security Constraints	77
	Defining Roles	77
	Selecting the Authentication Method	78
	Configuring the UserDatabaseRealm	78
12	Configuring SSL	79
	General Information	80
	Creating Your Own Trust File	80
	Defining SSL Usage in the Configuration File	81
13	Logging	83
	General Information	84
	Name and Location of the Configuration File	84
	Invoking the Logging Configuration Page	84
	Overview of Options for the Output File	86

Preface

This documentation is organized under the following headings:

Introduction	What is the Natural Web I/O Interface?
Installing and Configuring the Natural Web I/O Interface Server	How to install and configure the Natural Web I/O Interface server in a Windows environment.
Installing the Natural Web I/O Interface Client	How to install the Natural Web I/O Interface client on an application server or in a servlet container so that it can be used with the Natural Web I/O Interface server.
Configuring the Client	How to define the information that is to appear in the logon page.



Note: This documentation only explains how to install the Natural Web I/O Interface server in a Windows environment. For information on how to install it in a mainframe, UNIX or OpenVMS environment, see the Natural documentation for the appropriate platform.

I Introduction

1 Introduction

▪ What is the Natural Web I/O Interface?	4
▪ Components of the Natural Web I/O Interface	4
▪ Executing a Natural Application in a Web Browser	5
▪ Client-Server Compatibility	6
▪ Terminology	6
▪ Differences in a SPoD Development Environment	7
▪ Restrictions When Using the Natural Web I/O Interface with Natural Applications	7
▪ Differences between the Natural Web I/O Interface Client and Terminal Emulation	9

This chapter describes the purpose and the functions of the Natural Web I/O Interface.



Note: This introduction mainly describes how the Natural Web I/O Interface works in a runtime (production) environment. The section *Differences in a SPoD Development Environment* briefly explains the special version that is used in a SPoD development environment.

What is the Natural Web I/O Interface?

The Natural Web I/O Interface is used to execute Natural applications in a web browser. It fully supports the following:

- The display and input of Unicode characters. See *Unicode Input/Output Handling in Natural Applications* in the *Unicode and Code Page Support* documentation.
- Rich internet applications developed with Natural for Ajax.

Components of the Natural Web I/O Interface

The Natural Web I/O Interface consists of a server and a client.

Server

The Natural Web I/O Interface server enables you to use a browser as the I/O device for Natural applications. The server does the user authentication, creates the Natural session and handles the I/O between Natural and the client. The Natural Web I/O Interface server is installed on the same machine as the Natural application.

Client

The client handles the communication between the user's web browser and the Natural Web I/O Interface server. It converts the output from the Natural application to web pages, and returns the user input to Natural.

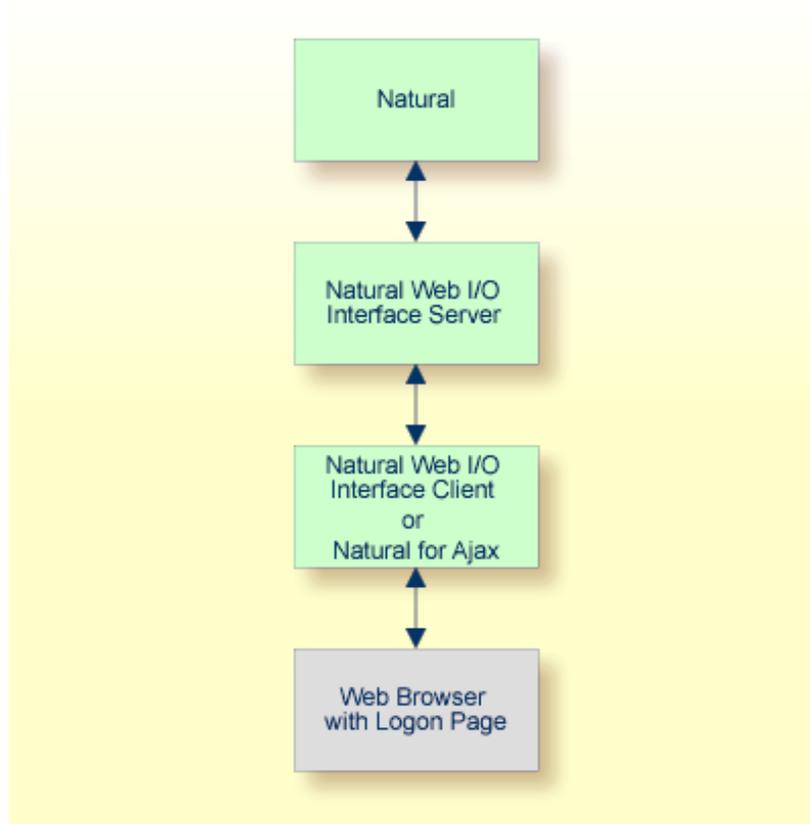
Two types of client are supported:

- Natural Web I/O Interface client for displaying character-based applications in the web browser. Maps with GUI controls are not supported in this case.
- Natural for Ajax for displaying rich internet applications in the web browser. For further information on this type of client, see the Natural for Ajax documentation.

The client is installed on a web/application server. This can be done on any machine in the network.

Executing a Natural Application in a Web Browser

The Natural Web I/O Interface receives data from a Natural application and delivers web pages to the user's web browser. This is illustrated in the following graphic:



The communication steps for executing a Natural application in the web browser are:

1. The user enters the address (URL) of a logon page in the web browser. The client then displays the logon page in the web browser.



Note: For information on how to invoke and configure the logon page, see [Configuring the Client](#).

2. The user enters all required information for starting a Natural application into the logon page. This information is sent to the client.
3. The client asks the Natural Web I/O Interface server to start the requested Natural application for this user.
4. The Natural Web I/O Interface server checks the supplied user ID and password, creates a Natural session for the user and starts the Natural application.

5. The Natural application returns the first application screen which is then transferred via the Natural Web I/O Interface server to the client and finally as a web page to the web browser.

Different web browsers are supported. Note that cookies and JavaScript must be enabled in the web browser. For a list of the currently supported web browsers, see the browser prerequisites for the type of client that you are using.

Client-Server Compatibility

The following rules apply:

- The Natural Web I/O Interface server can work with any client that has the same or a higher protocol version.

If the server detects that the client is using a version that is lower than the server version, the server replies that the client is too old and the connection is closed.

- The client can work with any server that has the same or a lower protocol version.

If the client detects that the server is using a version that is lower than the client version, the client switches to the server version. However, new client functionality is not supported in this case.

- The Natural Web I/O Interface server must have the same protocol version as the Natural process that is started by the server. If Natural detects that the server is using a different protocol version, an error message is sent to the user and the connection is closed.

Terminology

On the different Natural platforms for which the Natural Web I/O Interface is supported, different techniques are used for implementing the server part of the Natural Web I/O Interface. On Natural for UNIX and Natural for OpenVMS, it is implemented as a daemon. On Natural for Windows, it is implemented as a service. On the mainframe, it is implemented as a server. In this documentation, the general term “server” is therefore used for all different kinds of implementation.

Differences in a SPoD Development Environment

The previous sections of this introduction have described how the Natural Web I/O Interface works in a runtime (production) environment. This section briefly explains the differences in a SPoD development environment.

A special version of the Natural Web I/O Interface is used when working in a remote development environment with Natural for Windows (SPoD). In this case, the Natural Web I/O Interface is an integrated component which does not require a separate installation. The server is part of the Natural Development Server (NDV), and the client is part of Natural Studio. Other than in the runtime environment, the screen is not displayed in a browser but in a normal window. Rich GUI pages created by Natural for Ajax are not supported in the development environment.

It is important that I/O via the Natural Web I/O Interface has been enabled on the Natural host. Otherwise, the Natural Web I/O Interface cannot be invoked. See also *Unicode Input/Output Handling in Natural Applications* in the *Unicode and Code Page Support* documentation.

Restrictions When Using the Natural Web I/O Interface with Natural Applications

There are several restrictions when using the Natural Web I/O Interface with Natural applications on UNIX, OpenVMS, mainframe or Windows hosts.



Note: The term “application” refers to application software. It does not refer to system software or software for development.

The following restrictions apply:

- **GUI controls**
GUI controls are not supported: dialogs, buttons, radio buttons, list boxes, list views, check boxes etc. The Natural Web I/O Interface only supports Natural applications developed without GUI controls.
- **File transfer**
File transfer (for example, with the `DOWNLOAD` statement) is not supported by the Natural Web I/O Interface.
- **Runtime errors**
This restriction applies to older Natural versions on UNIX and Windows. As of version 6.3.3, this restriction no longer applies.

Runtime errors in Natural applications are not handled by the Natural Web I/O Interface. This leads to a loss of the session. Bypass: use the Natural system variable *ERROR-TA to handle the error. Sample Natural error transaction:

```
DEFINE DATA
LOCAL
1 ERR_INFO
  2 ERR_NR(N5)
  2 ERR_LINE(N4)
  2 ERR_STAT(A1)
  2 ERR_PNAM(A8)
  2 ERR_LEVEL(N2)
END-DEFINE
INPUT ERR_INFO
DISPLAY ERR_INFO
TERMINATE
END
```

■ **Terminal commands**

Terminal commands are not supported. They do not work when entered in the Natural Web I/O Interface client.

■ **Natural system variable *INIT-ID**

When using the Natural Web I/O Interface client with Natural applications on UNIX, OpenVMS, mainframe or Windows hosts, the Natural system variable *INIT-ID will not be filled with a value for the terminal type. On UNIX, OpenVMS and Windows, it will contain the value "notty". On mainframes, it will contain a session ID that is unique on that server.

The following restrictions apply to Natural on UNIX, OpenVMS and Windows hosts (the mainframe does not have these restrictions):

■ **Return to the Natural main screen**

You must not use Natural applications that return to the Natural main screen as this leads to wrong screen display and a loss of the session.

■ **Natural editors and utilities**

You must not use Natural utilities such as SYSMAIN or SYSDDM and editors such as the program editor as this leads to wrong screen display and a loss of the session.

■ **Natural system commands**

You must not use any Natural system command such as CATALL, FIND, GLOBALS, HELP, KEY, LIST, RETURN, SCAN, SETUP or XREF as this leads to wrong screen display and a loss of the session.

Differences between the Natural Web I/O Interface Client and Terminal Emulation

The Natural Web I/O Interface client runs as an HTML terminal emulator inside a browser control. The look and feel of the Natural Web I/O Interface client display is quite similar to that of the regular terminal (emulation), but there are some differences due to browser functionality:

- A double-click with the mouse pointer on any field simulates the `ENTER` key.
- It is not possible to position the cursor outside the range of input and output fields.
- The cursor can be moved with the left and right arrow keys within one input field. It is also possible to jump from one input field to another using the left, right, up and down arrow keys.
- The insert mode can be switched on and off using the `INSERT` key.
- For Unicode character sets (type `U`; for example, Chinese), one character may require more space than an ordinary alphanumeric character, because the Unicode character representation is proportional. The application design must take this into account, because Natural is based on characters with fixed width. For input fields it is possible to scroll within the field, but for output fields there may not be sufficient space to display the Unicode characters. The display length for a field can be controlled by the session parameter `DL`.
- Type-ahead mode is not supported.
- Paste in overwrite mode is not supported.
- Key schemes are fixed; keys such as the right `CTRL` key and the `ENTER` key on the numeric pad are no longer definable.
- Screen update is slower since the complete screen is sent rather than updates.
- The blink attribute is not supported in Internet Explorer.
- The keys `PF1` through `PF12` are simulated by the key combinations `F1` through `F12`.
- The keys `PF13` through `PF24` are simulated by the key combinations `SHIFT+F1` through `SHIFT+F12`.
- The keys `PF25` through `PF36` are simulated by the key combinations `CTRL+F1` through `CTRL+F12`.
- The keys `PF37` through `PF48` are simulated by the key combinations `ALT+F1` through `ALT+F12`.
- The program attention keys (`PA1`, `PA2` and `PA3`) are simulated by the key combinations `CTRL+SHIFT+F1`, `CTRL+SHIFT+F2`, `CTRL+SHIFT+F3`.
- The clear key is simulated by `CTRL+SHIFT+F4`.

IBM Mainframes Only

- The terminal screen size is controlled by the Natural profile parameter `TMODEL`. The default setting `TMODEL=0` means 24 lines and 80 columns.
- There is no `ATTN` (attention interrupt) key, no `RESET` key and no `EEOF` (erase end of file) key.

VT Only

- The I/O occurs in block mode. Therefore, the Natural program will only react when a function key is pressed.

II Installing and Configuring the Natural Web I/O Interface Server

2 Installing and Configuring the Natural Web I/O Interface

Server

- Installing the Natural Web I/O Interface Service 14
- Configuring the Natural Web I/O Interface Service on Windows 14
- Logging Information 19
- SSL Support 20

On Windows, the server part of the Natural Web I/O Interface runs in the background as a so-called service.

Installing the Natural Web I/O Interface Service

The Natural Web I/O Interface service is installed with Natural for Windows if the corresponding option is set during the installation. See the *Installation* documentation for further information.

Configuring the Natural Web I/O Interface Service on Windows

The Natural Web I/O Interface service is installed as a Windows service. In the Windows Services management console, the service is listed as **Software AG Natural *n.n* Web I/O Service**. It is started automatically at Windows startup. By default, the service listens on the port number 2900. You can choose a different port number during installation.

The configuration of the Natural Web I/O Interface service is stored in the Windows registry. If you do not want to use the default values with which the Natural Web I/O Interface service has been installed, you can modify the service configuration as described below to meet your requirements.

The following topics are covered below:

- [Natural Web I/O Interface Service Commands](#)
- [Example: Starting Natural Web I/O Interface Service with Your Own Configuration](#)
- [Batch File for Starting Natural](#)
- [Environment Variables](#)

Natural Web I/O Interface Service Commands

The program *nwosvcd.exe*, which is stored in the Natural *bin* directory, is used to execute the service commands. The following service commands can be specified in the **Command Prompt** window of Windows:

Command	Description
<code>nwosvcd -help</code>	Shows all available Natural Web I/O Interface service commands and subcommands in a message box.
<code>nwosvcd -install</code>	Installs the Natural Web I/O Interface service. The service is installed with the startup type Automatic .

Command	Description
	If you want to change the startup type, you must open the Windows Services management console. Open the properties for Software AG Natural <i>n.n</i> Web I/O Service and select the desired startup type.
nwosvcd -start	Starts the Natural Web I/O Interface service from the system.
nwosvcd -stop	Stops the Natural Web I/O Interface service from the system.
nwosvcd -remove	Removes the Natural Web I/O Interface service from the system.
nwosvcd -console	<p>Runs the Natural Web I/O Interface service as a Windows console application. In this case, an icon appears in the notification area of the taskbar. When you invoke the context menu for this icon, the following commands are provided:</p> <ul style="list-style-type: none"> ■ About Shows the version of the Natural Web I/O Interface service. ■ Exit Invokes a dialog box, asking whether you want to stop the Natural Web I/O Interface service or not.
nwosvcd -show	Shows the configuration of the Natural Web I/O Interface service.
nwosvcd -config keys	<p>Changes the configuration of the Natural Web I/O Interface service. The following keys can be specified:</p> <p>-host=<i>hostname</i> The host name used.</p> <p>-port=<i>nnnn</i> The port number.</p> <p>-userexit1=<i>pathname</i> The library that is defined by <code>userexit1</code> must contain the following function:</p> <pre>int nwo_CheckUsernameAndPassword(const char *pUsername, const char *pPassword, const char *pNewPassword, char *pErrorMessage, HANDLE *pHandleUser)</pre> <p>If the key <code>userexit1</code> is configured, the function <code>nwo_CheckUsernameAndPassword</code> is responsible for checking the user name and password. If a new password is received, <code>user exit 1</code> is also responsible for changing the password.</p> <p>In the case of an error, the return code of the function must be "0"; in this case, the <code>pErrorMessage</code> is returned to the client. When user name and password are correct, the return code must be a value other than "0".</p> <p>-userexit2=<i>pathname</i> The library that is defined by <code>userexit2</code> must contain the following functions:</p> <ul style="list-style-type: none"> ■ <code>int nwo_Messages(int *iNumberOfMessages, char *pMessage[])</code> <p>If the key <code>userexit2</code> is configured, the function <code>nwo_Messages</code> is called when a new client is accepted and the messages that are returned by this function are sent to the client. <code>User exit 2</code> may be used, for example, to send a message such</p>

Command	Description
	<p>as the following: "For maintenance reasons, the Natural application XXXXX will be down next monday, from 18:00 until 19:00".</p> <p>In the case of an error, the return code of the function must be "0".</p> <p>After the function <code>nwo_Messages</code> has been called, the function <code>nwo_FreeMessages</code> is called.</p> <ul style="list-style-type: none"> ■ <code>int nwo_FreeMessages(int iNumberOfMessages, char *pMessage[])</code> <p><code>iNumberOfMessages</code>: Number of messages.</p> <p><code>pMessage</code>: Array of messages.</p> <p>If the key <code>userexit2</code> is defined, the function <code>nwo_FreeMessages</code> is called to free any resources (normally memory) allocated in the function <code>nwo_Messages</code>.</p> <p>In the case of an error, the return code of the function must be "0".</p> <p><code>-logging=option</code> The amount of logging information that is to be reported. One of the following options can be specified:</p> <ul style="list-style-type: none"> E for errors. W for warnings. I for information. <p>See also Logging Information.</p> <p><code>-ssl=[yes no]</code> Defines whether the SSL protocol is to be used. See also SSL Support.</p> <p>To remove any user exits from the configuration, enter the following command:</p> <pre>nwosvcd -config -userexit1=</pre> <p>Once the configuration was changed, the Natural Web I/O Interface service must be restarted.</p>



Note: The service commands are not case-sensitive. They can be specified in both upper-case and lower-case.

Example: Starting Natural Web I/O Interface Service with Your Own Configuration

This example explains how to create a new configuration with the port number 3344 and the user exit 1 `C:\Program Files\My Company Name\User Exit1\libuserexit1.dll`.

➤ To start the Natural Web I/O Interface service with your own configuration

- 1 Open the Windows Services management console and stop the Natural Web I/O Interface service.
- 2 Invoke the **Command Prompt** window of Windows.
- 3 Go to the Natural *bin* directory which contains the file `nwosvcd.exe`.
- 4 Enter the following command to configure a new port number and a new user exit 1:

```
nwosvcd -config -port=3344 -userexit1="C:\Program Files\My Company Name\User Exit1\libuserexit1.dll"
```

- 5 Go back to the Windows Services management console and start the Natural Web I/O Interface service.

Batch File for Starting Natural

In order to start a Natural session, the Natural Web I/O Interface service executes a batch file. The batch file prepares the environment for the Natural session and eventually starts Natural. It must therefore contain all environment settings needed to run the Natural session.

The batch file receives certain parameters from the Natural Web I/O Interface client. The parameters can either be evaluated by the batch file itself or passed on to Natural. A client who wants to start a Natural session can specify the batch file to be used. The section [Configuring the Client](#) describes where and how this is done.

A sample batch file with the name `nwo.bat` is contained in the Natural *bin* directory. It has the following content:

```
@echo off
set IPAddress=""
set ClientID=""
set Argument3=""
set Argument4=""
set NaturalParameters=""

if not (%1)==(null) set IPAddress=%~1

if not (%2)==(null) set ClientID=%~2

if not (%3)==(null) set Argument3=%~3
```

```

if not (%4)==(null) set Argument4=%~4

if not (%5)==(null) set NaturalParameters=%~5

echo IPAddress=%IPAddress% >nwo.log
echo ClientID=%ClientID% >>nwo.log
echo Argument3=%Argument3% >>nwo.log
echo Argument4=%Argument4% >>nwo.log
echo NaturalParameters=%NaturalParameters% >>nwo.log

natrt.exe %NaturalParameters% etid=$$

```

You can create your own batch files to start up Natural sessions. If you do so, you should make sure - as in the above sample batch file - to start Natural with *natrt.exe*. The above sample requires that the batch file is contained in the Natural *bin* directory. If this is not the case, you must specify an appropriate absolute or relative path to *natrt.exe*.

In order to refer to a specific batch file in the Natural Web I/O Interface client, you can specify either an absolute path or a relative path. If you specify a relative path, the batch file is searched according to the following rules:

1. In the directory where *nwosvcd.exe* is located. This is the Natural *bin* directory.
2. In the system directory. This is the directory that the system function `GetSystemDirectory` would deliver.
3. In the Windows directory. This is the directory that the system function `GetWindowsDirectory` would deliver.
4. In the directories that are listed in the `PATH` environment variable.

Arguments

The batch file will receive the following arguments:

Order	Argument	Description
1	IPAddress	The client IP address from where the session is opened. Note: If there is a proxy, this will not be the IP address of the client workstation. Instead, it will be the IP address of the proxy.
2	ClientId	The user name from the logon page is passed as the client ID.
3	Argument3	Reserved for future use.
4	Argument4	Reserved for future use.
5	NaturalParameters	These can be any Natural parameters. The parameters are either defined in the configuration file for the session, or they are entered in the logon page. The following is an example of the corresponding entry in the configuration file:

Order	Argument	Description
		<pre><natural_parameter>parm=nwoparm\ stack=(logon\ mylib;start-program;fin)<natural_parameter></pre> <p>The language that is selected in the logon page is added as the first element to the Natural parameters in the form "ulang=x".</p>

Arguments 1 and 2 can be used to audit the client, to allow to run an application from a specific PC (identifying the IP address), to build statistics, to do special actions, etc.

Environment Variables

In the batch file, several environment variables can be set for the Natural session that is started by the service:

NWO_ENABLE_ACK=["YES"|"NO"]

This environment variable is used for asynchronous screens (SET CONTROL N).

YES When asynchronous screens are sent to the client, Natural will wait to receive an ACK package before the next screen can be sent.

NO No waiting between asynchronous screens. Default value.

NWO_TIMEOUT=[*number-of-seconds*]

The maximum time, in seconds, that Natural waits to receive any input from the client before it closes the session. If the number of seconds is "0", Natural waits infinitely (no timeout). The default value is "0".

Error NAT5466 is returned at timeout. In Natural, the application can handle this error and decide how to continue or terminate.

Logging Information

The logging information system reports errors, warnings and/or session information, depending on the option that has been defined with the following **Natural Web I/O Interface service command**:

```
nwosvcd -config -logging=option
```

option can be one of the following:

Option	Description
E	<p>Error.</p> <p>When this option is specified, the Natural Web I/O Interface service reports only errors.</p> <p>In the case of an error, the service usually exits immediately.</p>
W	<p>Warning.</p> <p>When this option is specified, the Natural Web I/O Interface service reports errors and warnings for uncritical errors.</p> <p>In the case of a warning, the service continues to run.</p>
I	<p>Information.</p> <p>When this option is specified, the Natural Web I/O Interface service reports errors, warnings and information.</p> <p>The information messages allow to check the session parameters, IP address, etc.</p>

Help information, for example, on how to run, configure and install the Natural Web I/O Interface service is always provided. The messages which inform you when the service has been started or stopped are also part of the help information.

To find out which logging option is currently active, enter the following Natural Web I/O Interface service command:

```
nwosvcd -show
```

The logging messages are saved in the Windows Event Viewer.

SSL Support

SSL is used for a secure connection between the Natural Web I/O Interface server and the Natural Web I/O Interface client or Natural for Ajax. Server authentication cannot be switched off. A certificate and a private key is always required on the server.

To establish an SSL connection, you have to proceed as described in the following topics:

- [Creating an SSL Certificate and a Private Key](#)
- [Configuring the Service](#)

- [Configuring the Client](#)

Creating an SSL Certificate and a Private Key

To create and use an SSL certificate and a private key on the server, proceed as described below.

1. Adapt the example configuration file *openssl.cnf* to your needs.

 **Note:** *openssl.cnf* and *openssl* are delivered in *<install-dir>\bin*.

2. Set the environment variable so that it points to the file *openssl.cnf*:

```
set OPENSSL_CONF=<install-dir>\bin\openssl.cnf
```

3. Generate a certificate signing request:

```
openssl req -new > server.cert.csr
```

4. Generate a private RSA key:

```
openssl rsa -in privkey.pem -out server.cert.key
```

5. Generate a self-signed certificate:

```
openssl x509 -in server.cert.csr -out server.cert.crt -req -signkey ↵  
server.cert.key -days 365
```

It is important that the name of the generated certificate is *server.cert.crt* and that the name of the generated private key is *server.cert.key*.

 **Note:** The certificate can be self-signed or it can be signed by a CA (Certificate Authority) such as VeriSign.

6. Put the generated files into the same directory as the scripts which start the Natural Web I/O Interface server.

Configuring the Service

After you have created an SSL certificate and a private key as described above, proceed as follows:

1. Change the configuration of the Natural Web I/O Interface service using the following command:

```
nwosvcd -config -ssl yes
```

2. Restart the Natural Web I/O Interface service.

See also [Configuring the Natural Web I/O Interface Service on Windows](#).

Configuring the Client

After you have configured the service as described above, you have to import the generated *server.cert.crt* file to a truststore on the client. For information on how to do this for the Natural Web I/O Interface client, see [Configuring SSL](#). If you are using Natural for Ajax as the client, see [Configuring SSL](#) in the Natural for Ajax documentation.

III

Installing the Natural Web I/O Interface Client

This part explains how to install the Natural Web I/O Interface client on Tomcat so that it can be used with the server part of the Natural Web I/O Interface that is running in a Natural for Mainframes, Natural for UNIX, Natural for OpenVMS or Natural for Windows runtime environment.

The following topics are covered:

Prerequisites

[Installing the Natural Web I/O Interface Client on Apache Tomcat](#)

[Migrating the Natural Web I/O Interface Client from IIS to Apache Tomcat](#)

3 Prerequisites

- Servlet Container 26
- Natural for Mainframes 26
- Natural for UNIX 26
- Natural for OpenVMS 27
- Natural for Windows 27
- Browser Prerequisites 27

Servlet Container

The following servlet container is supported. The servlet container is not delivered with the Natural Web I/O Interface. It can be obtained from the location indicated below, according to its license terms.

- **Apache Tomcat 7 and 8**
See <http://tomcat.apache.org/>.

Natural for Mainframes

If you want to use the Natural Web I/O Interface client with Natural for Mainframes, the following must be installed:

- Natural for Mainframes Version 8.2.5 or above, and
- the Natural Web I/O Interface server.

For detailed information, see:

- the *Installation* documentation for the different operating systems which is provided for Natural for Mainframes;
- the section *Installing and Configuring the Natural Web I/O Interface Server* in the version of this *Natural Web I/O Interface* documentation which is provided for Natural for Mainframes.

Natural for UNIX

If you want to use the Natural Web I/O Interface client with Natural for UNIX, the following must be installed:

- Natural for UNIX Version 8.3 or above, and
- the Natural Web I/O Interface server (which is implemented as a daemon).

For detailed information, see:

- the *Installation* documentation which is provided for Natural for UNIX;
- the section *Installing and Configuring the Natural Web I/O Interface Server* in the version of this *Natural Web I/O Interface* documentation which is provided for Natural for UNIX.

Natural for OpenVMS

If you want to use the Natural Web I/O Interface client with Natural for OpenVMS, the following must be installed:

- Natural for OpenVMS Version 6.3.4 or above, and
- the Natural Web I/O Interface server (which is implemented as a daemon).

For detailed information, see:

- the *Installation* documentation which is provided for Natural for OpenVMS;
- the section *Installing and Configuring the Natural Web I/O Interface Server* in the version of this *Natural Web I/O Interface* documentation which is provided for Natural for OpenVMS.

Natural for Windows

If you want to use the Natural Web I/O Interface client with Natural for Windows, the following must be installed:

- Natural for Windows Version 8.3 or above, and
- the Natural Web I/O Interface server (which is implemented as a service).

For detailed information, see:

- the *Installation* documentation which is provided for Natural for Windows;
- the section *Installing and Configuring the Natural Web I/O Interface Server* in the version of this *Natural Web I/O Interface* documentation which is provided for Natural for Windows.

Browser Prerequisites

Supported browsers in this version are:

Internet Explorer 11 ⁽¹⁾

Microsoft Edge

Mozilla Firefox Extended Support Release 38 and 45 ⁽²⁾

Google Chrome ⁽³⁾

Notes:

⁽¹⁾ The Natural Web I/O Interface client Version 1.3.15 is the last version that supports Internet Explorer 8.

⁽²⁾ Only the Extended Support Releases of Mozilla Firefox are explicitly supported. Due to frequent upgrades of the Mozilla Firefox consumer release, the compatibility of the Natural Web I/O Interface client with future versions of Mozilla Firefox cannot be fully guaranteed. Possible incompatibilities will be removed during the regular maintenance process of the Natural Web I/O Interface client.

⁽³⁾ The Google Chrome support is based on Google Chrome Version 31. Due to frequent version upgrades of Google Chrome, compatibility of the Natural Web I/O Interface client with future versions of Google Chrome cannot be fully guaranteed. Possible incompatibilities will be removed during the regular maintenance process of the Natural Web I/O Interface client.



Important: Cookies and JavaScript must be enabled in the browser.

4 Installing the Natural Web I/O Interface Client on Apache

Tomcat

- Installation Steps 30
- Installation Verification 32

If you want to use the Natural Web I/O Interface client with Apache Tomcat, you must proceed as described in this chapter.

Installation Steps

The Natural Web I/O Interface client is installed using the Tomcat Manager.

The following is assumed:

- `<install-dir>` is the path to Software AG's main installation directory. By default, this is `C:\SoftwareAG` on Windows and `/opt/softwareag` on UNIX.
- `<host>` is the name of the machine on which Apache Tomcat is installed.
- `<port>` is the name of the port where Apache Tomcat is installed. In a default installation, this is port 8080.
- `<tomcat>` is the path to the directory in which Apache Tomcat is installed.

The following topics are covered below:

- [First-time Installation](#)
- [Update Installation](#)

First-time Installation

➤ To install the Natural Web I/O Interface client

- 1 Natural for Windows and UNIX: Copy the complete contents of the `<install-dir>/natural/INSTALL/nwoclient/j2ee/v<nnnn>/tomcat` directory to a directory of your choice on your hard disk.

Or:

Download the Natural Web I/O Interface client for Apache Tomcat from Empower (<https://empower.softwareag.com/>) and unzip the contents to a directory of your choice on your hard disk.

- 2 On UNIX platforms: Dearchive the TAR file using the following command:

```
tar -xvf nwonnnn.tar
```

- 3 Make sure that Apache Tomcat is running.
- 4 Open your web browser and enter the following URL:

```
http://<host>:<port>/manager/html
```

This opens the Tomcat Manager.

- 5 Deploy the web application file *natuniweb.war*:
 - Under **Select WAR file to upload** select the path to the file *natuniweb.war*.
 - Choose **Deploy**.
- 6 In the Tomcat Manager, look for the application **Natural Web I/O Interface Client** and choose **Reload**.

Update Installation

» To update the Natural Web I/O Interface client

- 1 Natural for Windows and UNIX: Copy the complete contents of the `<install-dir>/natural/INSTALL/nwoclient/j2ee/v<nnnn>/tomcat` directory to a directory of your choice on your hard disk.

Or:

Download the Natural Web I/O Interface client for Apache Tomcat from Empower (<https://empower.softwareag.com/>) and unzip the contents to a directory of your choice on your hard disk.

- 2 On UNIX platforms: Dearchive the TAR file using the following command:

```
tar -xvf nwonnnn.tar
```

- 3 Shut down Apache Tomcat.
- 4 Create a backup copy of your *sessions.xml* file, which is located in `<tomcat>/webapps/natuniweb/WEB-INF`.
- 5 Start Apache Tomcat.
- 6 Open your web browser and enter the following URL:

```
http://<host>:<port>/manager/html
```

This opens the Tomcat Manager.

- 7 Select *natuniweb.war* in the list of installed applications.
- 8 Choose **Undeploy**.
- 9 Deploy the new version of the Natural Web I/O Interface client as in a first-time installation.
- 10 Restore the *sessions.xml* file that you have backed up previously.

Installation Verification

It is assumed that `http://<host>:<port>` is the URL of your application server.

➤ To verify the installation

- Enter the following URL in your web browser:

```
http://<host>:<port>/natuniweb/natural.jsp
```

For example:

```
http://myhost:8080/natuniweb/natural.jsp
```

The Natural Web I/O Interface client is now started in your browser. The entries which appear in the resulting logon page depend on the settings in your configuration file. For further information, see [Configuring the Client](#).

5 Migrating the Natural Web I/O Interface Client from IIS to Apache Tomcat

- Before You Install the Natural Web I/O Interface Client 34
- Installing the Natural Web I/O Interface Client on Apache Tomcat 35
- Configuring the Natural Web I/O Interface Client on Apache Tomcat 35

Microsoft Internet Information Services (IIS) is no longer supported. If you are currently using the Natural Web I/O Interface client on IIS, you have to move to Apache Tomcat.



Note: JBoss Application Server and Oracle GlassFish Server are also no longer supported. If you are currently using one of these application servers, you also have to move to Apache Tomcat. In this case, however, you can reuse your previous settings (that is, the URL for logon page and the configuration file *sessions.xml*).

The most simple solution is to migrate the Natural Web I/O Interface client from IIS to Apache Tomcat. Therefore, this chapter gives IIS administrators a quick introduction to a Tomcat installation and describes the migration steps.

Before You Install the Natural Web I/O Interface Client

If Apache Tomcat is not yet installed, proceed as described in the topics below:

- [Installing Tomcat](#)
- [Installing Java](#)
- [Starting the Tomcat Server](#)

Installing Tomcat

Go to <http://tomcat.apache.org/download-60.cgi> and download Tomcat 6 as a zip file.

For Microsoft Windows users: download either the 32-bit or the 64-bit Windows zip file.

Unzip the downloaded zip file to a directory of your choice.

Installing Java

Tomcat is based on Java. Therefore, you have to make sure that a Java Runtime Environment (JRE) or a Java Development Kit (JDK) is installed. The version of the Java runtime should be at least Java 6 update 24. This is the minimum version that is required for the Natural Web I/O Interface client on Tomcat.

You can download the Java JRE or JDK from the Oracle website at <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

If Java is installed on your system, make sure that the environment variable `JAVA_HOME` is set to the Java home directory.

Starting the Tomcat Server

When Tomcat and the appropriate Java version have been installed, you can start Tomcat.

To start Tomcat, execute the *startup.bat* file from the *bin* directory of your Tomcat installation. To check whether Tomcat is running, enter the following URL:

```
http://localhost:8080
```

This should display Tomcat's default home page.

Installing the Natural Web I/O Interface Client on Apache Tomcat

When Apache Tomcat has been installed, install the Natural Web I/O Interface client as described in [Installing the Natural Web I/O Interface Client on Apache Tomcat](#).

Configuring the Natural Web I/O Interface Client on Apache Tomcat

When the Natural Web I/O Interface client has been installed, proceed as described in the topics below:

- [Invoking the Logon Page](#)
- [Changing the Tomcat HTTP Port](#)
- [Using the Settings from Your IIS Configuration File](#)
- [Using the Configuration Tool](#)
- [Protecting the Configuration Tool Against Unauthorized Access](#)
- [Displaying the Logon Page by Default](#)

Invoking the Logon Page

Enter the following URL to invoke the logon page (this is different from the URL that was used with IIS):

```
http://localhost:8080/natuniweb/natural.jsp
```

Changing the Tomcat HTTP Port

IIS usually runs on the default port 80. If you want Tomcat to work with the same port, edit the file *server.xml* which is located in Tomcat's *conf* subdirectory and then search for the following text:

```
<Connector port="8080" protocol="HTTP/1.1"
```

Change the port number so that it looks as follows:

```
<Connector port="80" protocol="HTTP/1.1"
```

Using the Settings from Your IIS Configuration File

With Tomcat, you can reuse the *settings.xml* configuration file of IIS, but you have to rename the file to *sessions.xml*. Proceed as follows:

1. Copy the *settings.xml* file from your IIS installation to the following directory of your Tomcat installation:

```
webapps/natuniweb/WEB-INF
```

2. Either rename the *sessions.xml* file which comes with the Natural Web I/O Interface client installation on Tomcat (for example, to *sessions-original.xml*) or delete it.
3. Rename the *settings.xml* file to *sessions.xml*.

Using the Configuration Tool

When the Natural Web I/O Interface client runs on Tomcat, it is no longer necessary to edit the configuration file manually. Instead, you can use the configuration tool. Using this tool has the advantage that it is not possible for you to create invalid XML code and thus damage the XML file. See [Using the Configuration Tool](#) for further information.

The IIS-specific entries in the renamed configuration file will be ignored. These are:

```
natural_parameter visible  
theme  
screen top  
screen left  
screen size  
screen pfkeypos
```

You can still edit the configuration file manually. However, this is no longer recommended.

Protecting the Configuration Tool Against Unauthorized Access

It is possible to protect the configuration tool against unauthorized access. See [Configuring Container-Managed Security](#) for detailed information.

For detailed information on the necessary realm configuration for Tomcat, see <http://tomcat.apache.org/tomcat-6.0-doc/realms-howto.html>.

Displaying the Logon Page by Default

When you enter the URL `http://localhost:8080/natuniweb`, Tomcat shows the default page of the Natural Web I/O Interface client which allows you to access either the **logon page** or the **configuration tool** of the Natural Web I/O Interface client.



Note: If you have defined a different port (for example, 80), make sure to use that port number in the URL.

This behavior is different from IIS which displays the logon page by default. If you also want Tomcat to display the logon page by default, edit the file `web.xml` which is located in Tomcat's `webapps\natuniweb\WEB-INF` directory and search for the following entry:

```
<welcome-file-list>
  <welcome-file>
    index.html
  </welcome-file>
</welcome-file-list>
```

Change the name of the welcome file to `natural.jsp` as shown in the following example:

```
<welcome-file-list>
  <welcome-file>
    natural.jsp
  </welcome-file>
</welcome-file-list>
```


IV

Configuring the Client

This part explains how to configure the Natural Web I/O Interface client so that it can be used in a Natural runtime environment. The following topics are covered:

About the Logon Page

Natural Client Session Configuration

Natural Client Configuration Tool

Natural Web I/O Style Sheets

Starting a Natural Application with a URL

Configuring Container-Managed Security

Configuring SSL

Logging

6 About the Logon Page

- Starting a Natural Application from the Logon Page 42
- Examples of Logon Pages 42
- Dynamically Changing the CICS Transaction Name when Starting a Session 43
- Specifying a Password in the Logon Page 44
- Changing the Password in the Logon Page 44
- Browser Restrictions 45

Starting a Natural Application from the Logon Page

When you start the Natural Web I/O Interface client in the browser, a logon page appears. The entries in this logon page depend on the settings in your configuration file (see [Natural Client Session Configuration](#)).

In order to start a Natural application from the logon page, you enter the following URL inside your browser:

```
http://<host>:<port>/natuniweb/natural.jsp
```

where *<host>* and *<port>* are the host name and port number of your application server.

Examples of Logon Pages

For each session definition that has been configured in the configuration file, an entry appears on the logon page. If the user selects the corresponding entry, only those parameters that were not preconfigured in the configuration file need to be specified in the logon page in order to start the application. Usually, you will preconfigure all connection parameters except user name and password.

The following example shows part of a logon page which results from a configuration file in which no special entries are defined for a session:

Enter connection details:

Session ID:

Host name:

Port number:

User name:

Password:

Application:

Natural parameters:

The following example shows part of a logon page which results from a configuration file in which many settings are already predefined (including user ID and password):

Enter connection details:

Session ID:

To log on to a session, you have to specify all required information in the logon page (for example, you select a session from the corresponding drop-down list box). When you choose the **Connect** button, the screen for the selected session appears.

Dynamically Changing the CICS Transaction Name when Starting a Session

The following description applies if you want to switch to a different CICS transaction on a mainframe.

You specify the CICS transaction name in the same text box in which you also specify the dynamic parameters for the Natural environment. So that the CICS transaction name can be evaluated, it is important that you specify it before any Natural parameters, using the following syntax:

```
<TA_NAME=name>
```

where *name* can be 1 to 4 characters long. This must be the name of an existing CICS transaction which applies to a CICS Adapter. It will override the transaction name which is currently defined in the configuration file for the CICS Adapter on the Natural Web I/O Interface server (NWO). Ask your administrator for further information.

Make sure to put the entire definition in angle brackets. When this definition is followed by a Natural parameter, insert a blank before the Natural parameter. Example:

```
<TA_NAME=NA82> STACK=(LOGON SYSCP)
```

If the specified CICS transaction name cannot be found, an error message occurs and the session cannot be started.



Note: The above definition for the CICS transaction name can also be specified in the **configuration tool**, in the same place where you also specify the Natural parameters, and together with the **URL parameter** natparam.

Specifying a Password in the Logon Page

The following information applies when the field for entering a password appears on the logon page. This field does not appear when a password has already been defined in the configuration file.

Under Windows, UNIX and OpenVMS, you always have to enter the operating system password, even if Natural Security is active.

On the mainframe, this is different: When Natural Security is not active, you have to enter the operating system password. When Natural Security is active, you have to enter the Natural Security password.

Changing the Password in the Logon Page

Currently, this functionality is only available for Natural for UNIX, Natural for OpenVMS and Natural for Windows.

The following information applies when the fields for entering a user ID and a password appear on the logon page. These fields do not appear when user ID and password have already been defined in the configuration file; in this case, it is not possible to change the password in the logon page.

When your password has expired, you are automatically asked for a new password. When you try to log on with your current password, an error message appears and input fields for changing the password are shown.

» To change the password

- 1 Choose the **Change password** button in the logon page.

The name of this button changes to **Don't change password** and the following two input fields are shown in the logon page:

- **New password**
- **Repeat new password**

- 2 Enter your user ID and your current password as usual.
- 3 Enter the new password in the two input fields.
- 4 Choose the **Connect** button to change the password.

Or:

If you do not want to change your password, choose the **Don't change password** button. The two input fields will then disappear.

Browser Restrictions

The browser's "Back" and "Forward" buttons do not work with the Natural Web I/O Interface client and should therefore not be used.

If you want to run two Natural sessions in parallel, you have to start a new instance of the browser (for example, by choosing the corresponding icon in the Quick Launch toolbar of Windows). You must not use the browser's "New Window" function. This would result in one session running in two browsers, which is not allowed.

7 Natural Client Session Configuration

- General Information 48
- Name and Location of the Configuration File 48

General Information

The configuration file is an XML file which is required to define the sessions that can be invoked from the logon page.

To edit the configuration file, you use the configuration tool. Using this tool has the advantage that it is not possible for you to create invalid XML code and thus damage the XML file. See [Natural Client Configuration Tool](#) for further information.

Name and Location of the Configuration File

The name of the configuration file is *sessions.xml*. It can be found in the *WEB-INF* directory.

`<tomcat-install-dir>/webapps/natuniweb/WEB-INF`

8

Natural Client Configuration Tool

▪ Invoking the Configuration Tool	50
▪ Session Configuration	51
▪ Logging Configuration	59
▪ Logon Page	59
▪ Logout	60

Invoking the Configuration Tool

The Natural Web I/O Interface client offers a configuration tool. The configuration tool is used to create the session configurations which are then available in the logon page. It can also be used for logging purposes in case of problems; however, this should only be done when requested by Software AG support.

The configuration tool is automatically installed when you install the Natural Web I/O Interface client.

➤ To invoke the configuration tool

- Enter the following URL in your browser:

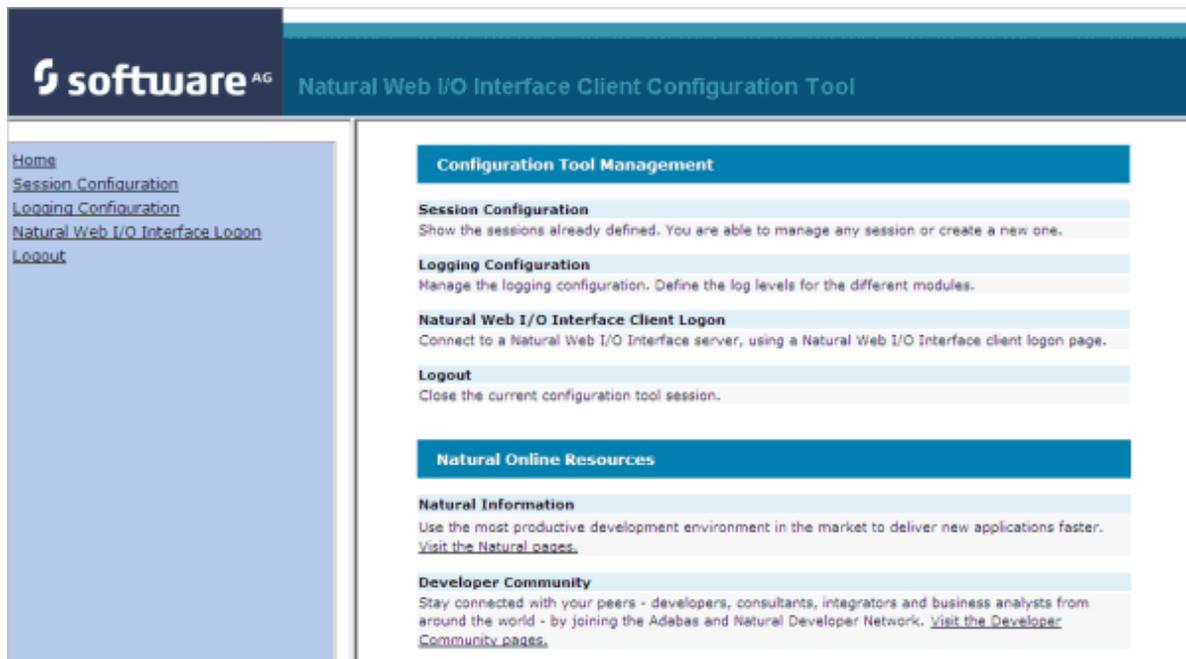
```
http://<host>:<port>/natuniweb/conf_index.jsp
```

where *<host>* and *<port>* are the host name and port number of your application server.



Note: You might wish to protect the configuration tool against unauthorized access. See [Configuring Container-Managed Security](#) for information on how to restrict the access to sensitive areas of the application server environment. If you have restricted access to the configuration tool, an authentication dialog appears. The appearance of this dialog depends on the authentication model you have chosen.

The configuration tool appears.



The configuration tool has two frames.

The home page of the configuration tool is initially shown in the right frame. It provides brief descriptions for the links provided in the left frame. It also provides links to several Software AG pages on the web.

When you have invoked a function (for example, when you are currently viewing the session configuration), you can always choose the **Home** link in the left frame to return to the home page of the configuration tool.

The functions that are invoked by the other links in the left frame are described below.

Session Configuration

This section explains how to manage the content of the configuration file for the sessions. It covers the following topics:

- [Invoking the Session Configuration Page](#)
- [Global Settings](#)
- [Adding a New Session](#)
- [Editing a Session](#)
- [Overview of Session Options](#)
- [Duplicating a Session](#)
- [Deleting a Session](#)
- [Adding a New User](#)
- [Saving the Configuration](#)

Invoking the Session Configuration Page

The content of the configuration file for the sessions is managed using the **Session Configuration** page.

➤ To invoke the Session Configuration page

- In the frame on the left, choose the **Session Configuration** link.

The **Session Configuration** page appears in the right frame. It shows the global settings and lists all sessions and users that are currently defined. For a session, some of the configuration file information is shown. Example:

Session Configuration

Global Settings

Last activity timeout (n seconds):

Trace directory:

SSL trust file path:

SSL trust file password:

Sessions

Session ID	Host Name	Port Number	Application	Natural Parameters	Edit	Duplicate	Delete
Connect to Natural					Edit	Duplicate	Delete
localtestserver	localhost	6640			Edit	Duplicate	Delete

Users

User ID	Edit	Duplicate	Delete
user1	Edit	Duplicate	Delete

Global Settings

The global settings apply for all defined sessions. You can define the following global settings in the configuration file:

Option	Description
Last activity timeout (n seconds)	The number of seconds that the client waits for the next user activity. When the defined number of seconds has been reached without user activity, the session is closed. The default is 3600 seconds.
Trace directory	<p>Optional. Location of a different trace directory.</p> <p>When a different trace directory is not defined, the trace files are written to the default trace directory. By default, the trace files are written to the directory which has been set by the Java property <code>java.io.tmpdir</code>. On Windows, this is normally the environment variable <code>TMP</code> for the user who started the application server. On UNIX, this is normally <code>/tmp</code> or <code>/var/tmp</code>.</p> <p>You can also set this property in the start script for the application server.</p>

Option	Description
	Tracing can be enabled individually for each session (see Overview of Session Options below). However, it should only be enabled when requested by Software AG support.
SSL trust file path	Optional. The path to your trust file. See Configuring SSL for further information.
SSL trust file password	If your trust file is password-protected, you have to specify the appropriate password. When you do not specify the password for a password-protected trust file, the trust file cannot be opened and it is thus not possible to open an SSL session. When your trust file is not password-protected, you should not specify a password.

Adding a New Session

You can add a new session to the configuration file.

➤ To add a new session

- 1 Choose the **Add New Session** button.

The **Edit Session** page appears.

- 2 Specify all required information as described below in the section [Overview of Session Options](#).
- 3 Choose the **OK** button to return to the **Session Configuration** page.

The new session is not yet available in the configuration file.

- 4 Choose the **Save Configuration** button to write the new session to the configuration file.

Editing a Session

You can edit any existing session in the configuration file.

➤ To edit a session

- 1 Choose the **Edit** link that is shown next to the session that you want to edit.

The **Edit Session** page appears.

- 2 Specify all required information as described below in the section [Overview of Session Options](#).
- 3 Choose the **OK** button to return to the **Session Configuration** page.

The modifications are not yet available in the configuration file.

- 4 Choose the **Save Configuration** button to write the modifications to the configuration file.

Overview of Session Options

The **Edit Session** page appears when you

- **add** a new session, or
- **edit** an existing session.

Example:

Edit Session

Session ID:	<input type="text"/>
Type:	Undefined <input type="text"/>
Host name:	<input type="text"/>
Port number:	<input type="text"/>
Use SSL	<input type="radio"/> Yes <input checked="" type="radio"/> No
User name:	<input type="text"/>
User name in upper case:	<input type="radio"/> Yes <input checked="" type="radio"/> No
Password:	<input type="text"/>
Application:	<input type="text"/>
Natural parameters:	<input type="text"/>
Double-click behavior:	Enter <input type="text"/>
Screen rows:	24 <input type="text"/>
Screen columns:	80 <input type="text"/>
Show function key numbers:	<input type="radio"/> Yes <input checked="" type="radio"/> No
Check for numeric input:	<input checked="" type="radio"/> Yes <input type="radio"/> No
Trace:	<input type="radio"/> Yes <input checked="" type="radio"/> No
Timeout (in seconds):	60 <input type="text"/>
Filler character:	<input type="text"/>

The **Edit Session** page provides the following options:

Option	Description
Session ID	Mandatory. A session name of your choice. On the logon page, the session name is provided in a drop-down list box.
Type	<p>The platform on which user ID and password are authenticated. You can select the required setting from the drop-down list box.</p> <ul style="list-style-type: none"> ■ Undefined Default. User ID and password can have a maximum of 32 characters. See also the description for Natural for Windows, UNIX or OpenVMS below. ■ Natural for Mainframes User ID and password can have a maximum of 8 characters. ■ Natural for Mainframes with Natural Security User ID and password can have a maximum of 8 characters. The user ID must comply with the Natural naming conventions for library names. ■ Natural for Windows, UNIX or OpenVMS User ID and password can have a maximum of 32 characters. When a domain is required, you have to specify it together with the user ID (in the form "<i>domain\user-ID</i>").
Host name	The name or TCP/IP address of the server on which Natural and the Natural Web I/O Interface server are running. When this is specified, the corresponding field does not appear on the logon page.
Port number	The TCP/IP port number on which the Natural Web I/O Interface server is listening. When this is specified, the corresponding field does not appear on the logon page.
Use SSL	<p>If set to Yes, a secure connection is established between the Natural Web I/O Interface client on the application server and the Natural Web I/O Interface server.</p> <p>Important: If you want to use SSL with Natural for Mainframes, one of the corresponding mainframe types must be selected; the type must not be Undefined or Natural for Windows, UNIX or OpenVMS. The other way around, if you want to use SSL with Natural for Windows, UNIX or OpenVMS, you must not select one of the mainframe types; the type may also be Undefined in this case.</p>
User name	Optional. A valid user ID for the current machine. When this is specified, the corresponding field does not appear on the logon page.
User name in upper case	If selected, the input field for the user ID is in upper-case mode.
Password	<p>Optional. A valid password for the above user ID.</p> <p>Under Windows, UNIX and OpenVMS, this is always the operating system password of the user, even if Natural Security is active.</p> <p>On the mainframe, this is different: When Natural Security is not active, this is the operating system password of the user. When Natural Security is active, this is the Natural Security password.</p>

Option	Description
	<p>When a password is specified, the corresponding field does not appear on the logon page. The configuration tool saves the password in encrypted form.</p>
<p>Application</p>	<ul style="list-style-type: none"> ■ Natural for Mainframes The name of the Natural program or a command sequence that starts your application as you would enter it on the NEXT prompt. Example: <code>TEST01 data1,data2</code> ■ Natural for UNIX The name of the UNIX shell script for starting the Natural application (a file similar to <i>nwo.sh</i>). ■ Natural for OpenVMS The name of the Natural image file (for example, <i>natural<version></i> or <i>natural<version>.exe</i>). ■ Natural for Windows The name of the Windows command file (<i>.bat</i>) for starting the Natural application. <p>When this is specified, the corresponding field does not appear on the logon page.</p>
<p>Natural parameters</p>	<p>Optional. Parameters for starting the Natural application. This can be stack parameters, a parameter file/module or other Natural-specific information.</p> <ul style="list-style-type: none"> ■ Natural for Mainframes Used to pass dynamic Natural profile parameters to the session, for example: <code>SYSPARM=(MYPARMS) STACK=(LOGON MYAPPL)</code> <p>Note: It is recommended to specify the Natural program that starts the application with the option Application instead of passing it with the profile parameter <code>STACK</code>.</p> <ul style="list-style-type: none"> ■ Natural for UNIX and Natural for Windows Used when the above shell script (UNIX) or command file (Windows) uses the parameter \$5 after "natural", for example: <code>PARM=MYPARM STACK=(LOGON MYLIB;MENU)</code> ■ Natural for OpenVMS Used for starting a Natural application, for example: <code>BP=BPnode-name NLDCHK WEBIO=ON "STACK=(LOGON SYSEXT;MENU)"</code>
<p>Double-click behavior</p>	<p>The key that is to be simulated when double-clicking an output field. By default, this is the ENTER key.</p> <p>It is possible to disable the double-click behavior, or to define a function key (PF1 through PF12).</p> <p>You can select the required setting from the drop-down list box.</p>

Option	Description
	<p>Tip: When context-sensitive help has been defined for the output fields, it may be useful to define PF1. The help function will then be invoked when the user double-clicks an output field.</p>
Screen rows	<p>The number of rows in the output window. Possible values: minimum 24, no upper limit. Default: 24.</p> <p>Not used by Natural for Mainframes which uses the profile parameter TMODEL instead.</p>
Screen columns	<p>The number of columns in the output window. Possible values: minimum 80, no upper limit. Default: 80.</p> <p>Not used by Natural for Mainframes which uses the profile parameter TMODEL instead.</p>
Show function key numbers	<p>If set to Yes, the PF key numbers are shown next to the PF keys.</p>
Trace	<p>Should only be set to Yes when requested by Software AG support.</p>
Check for numeric input	<p>If set to Yes (default), numeric input fields are validated. In this case, only the following characters are allowed in numeric input fields (in addition to the numbers "0" through "9"):</p> <p><i>blank</i> + (plus) - (minus) _ (underscore) , (comma) . (period) ? (question mark)</p> <p>If set to No, numeric input fields are not validated.</p>
Timeout (in seconds)	<p>The number of seconds that the client waits for a response after an updated page was sent to the Natural session. When the defined number of seconds has been reached without response, the session is closed. The default is 60 seconds. Normally, you need not change this value.</p>
Filler character	<p>Optional. The filler character that is to be removed from the input fields. An application can define, for example, an underscore (_) as the filler character. Trailing filler characters will be removed from the input fields, and leading filler characters will be replaced with blanks.</p>

Duplicating a Session

You can add a copy of any existing session to the configuration file.

➤ To duplicate a session

- 1 Choose the **Duplicate** link that is shown next to the session that you want to duplicate.

A new entry is shown at the bottom of the list of sessions. Its name is "Copy of *session-ID*".
The duplicated session is not yet available in the configuration file.
- 2 **Edit** and save the duplicated session as described above.

Deleting a Session

You can delete any existing session from the configuration file.

➤ To delete a session

- 1 Choose the **Delete** link that is shown next to the session that you want to delete.

The session is deleted from the list of sessions. It is not yet deleted in the configuration file.
- 2 Choose the **Save Configuration** button to delete the session from the configuration file.

Adding a New User

You can predefine Natural users and their passwords in the configuration file.

When a Natural page is opened with a URL that specifies a user in the URL parameter `natuser`, the specified user is matched against the list of users in the configuration file. When the specified user is defined in the configuration file, the corresponding password is used to authenticate the user when the Natural session is started. See also [Starting a Natural Application with a URL](#).

Example - when the following URL is used, the password defined for "user1" is used:

`http://myhost:8080/natuniweb/natural.jsp?natuser=user1...`

➤ To add a new user

- 1 Choose the **Add New User** button.

The **Edit User** page appears.
- 2 Specify a user name and password
- 3 Choose the **OK** button to return to the **Session Configuration** page.

The new user is not yet available in the configuration file.

- 4 Choose the **Save Configuration** button to write the new user to the configuration file.



Note: You edit, duplicate and delete a user in the same way as a session (see the corresponding descriptions above).

Saving the Configuration

When you choose the **Save Configuration** button, all of your changes are written to the configuration file. The server picks up the new settings automatically the next time it reads data from the configuration file.



Caution: If you do not choose the **Save Configuration** button but log out instead or leave the configuration tool by entering another URL, the new settings are not written to the configuration file.

Logging Configuration

The content of the configuration file for logging is managed using the **Logging Configuration** page. See the section [Logging](#) for detailed information.

Logon Page

The configuration tool provides the following link in the left frame:

- **Natural Web I/O Interface Logon**

This link opens the logon page in the right frame.

The logon page uses the current settings in the configuration file. When you select a session from the drop-down list box, you can check whether the connection details are shown as desired. If not, you can go back to the session configuration and modify the settings of the corresponding session.

See also [About the Logon Page](#).

Logout

When the configuration tool is protected against unauthorized access and you log out of the configuration tool, you make sure that no other user can change the client configuration when you leave your PC unattended for a while.

> To log out

- In the frame on the left, choose the **Logout** link.

When the configuration tool is protected against unauthorized access, the authentication dialog is shown again.

When it is not protected, the home page is shown again.

9 Natural Web I/O Style Sheets

- Name and Location of the Style Sheets 62
- Editing the Style Sheets 62
- Modifying the Position of the Main Output and of the PF Keys 62
- Modifying the Font Size 64
- Modifying the Font Type 65
- Defining Underlined and Blinking Text 65
- Defining Italic Text 66
- Defining Bold Text 66
- Defining Different Styles for Output Fields 67
- Modifying the Natural Windows 67
- Modifying the Message Line 68
- Modifying the Background Color 68
- Modifying the Color Attributes 69
- Modifying the Style of the PF Key Buttons 70
- XSLT Files 70

Name and Location of the Style Sheets

Several aspects on a page (such as font, font style or color) are controlled by a style sheet (CSS file).

The Natural Web I/O Interface client is delivered with the style sheet `3270.css`, which is located in:

`../natuniapp.ear/natuniweb.war/resources`



Note: For more information on style sheets, see <http://www.w3.org/Style/CSS/>.

Editing the Style Sheets

It is recommended that you have a basic understanding of CSS files.

You can edit the predefined style sheets or create your own style sheets.

It is recommended that you work with backup copies. When a problem occurs with your style sheet, you can thus always revert to the original state.

To see your changes in the browser, you have to

1. delete the browser's cache, and
2. restart the session.

Modifying the Position of the Main Output and of the PF Keys

Applies when only the named PF keys are displayed. This feature cannot be used when all PF keys are displayed, since they are always displayed at the same position. See also [Overview of Session Options](#).

The following elements are available:

Element Name	Description
<code>.mainlayer</code>	Controls the position of the main output in the output window. Used for languages that are written from left-to-right (LTR).
<code>.mainlayer_rtl</code>	Controls the position of the main output in the output window. Used for languages that are written from right-to-left (RTL).
<code>.pfkeydiv</code>	Controls the position of the PF keys in the output window. Used for languages that are written from left-to-right (LTR).

Element Name	Description
.pfkeydiv_rtl	Controls the position of the PF keys in the output window. Used for languages that are written from right-to-left (RTL).

The *_rtl elements are only used if Natural sends the web I/O screen with a right-to-left flag (SET CONTROL 'VON'). In the browser, the screen elements are then shown on the right side (instead of the left side).

For web I/O in applications where only the left-to-right orientation is used, the *_rtl elements are not required.

If the PF keys are to appear at the bottom, define the elements as shown in the following example:

```

/* Defines the main screen position */
.mainlayer {
    top: 5px;
    left: 0px;
    height: 550px;
}

/* Defines the main screen position for right-to-left */
.mainlayer_rtl{
    top: 5px;
    right: 30px;
    height: 550px;
}

/* Defines the PF keys screen position */
.pfkeydiv {
    height: 70px;
    left: 0px;
    top: 580px;
    width: 100%;
}

/* Defines the PF keys screen position for right-to-left */
.pfkeydiv_rtl {
    height: 70px;
    right: 30px;
    top: 580px;
    width: 100%;
}

```

Modifying the Font Size

Depending on the screen resolution, one of the following style sheets for defining the font size is used in addition to the default style sheet:

- *model2.css*
- *model3.css*
- *model4.css*
- *model5.css*

These style sheets are located in the *tmodels* subdirectory of the *resources* directory in which all style sheets are located.

Depending on what comes closest to the standard 3270 screen model, the corresponding style sheet from the *tmodels* subdirectory is automatically used. It is selected according to the following criteria:

Standard 3270 Screen Model	Criteria	Style Sheet
Model 2 (80x24)	30 rows or less.	<i>model2.css</i>
Model 3 (80x32)	Between 31 and 40 rows.	<i>model3.css</i>
Model 4 (80x43)	41 rows or more.	<i>model4.css</i>
Model 5 (132x27)	30 rows or less, and more than 100 columns.	<i>model5.css</i>

The font sizes in the above style sheets can be adjusted. Example for *model4.css*:

```
body {  
    font-size: 10px;  
}
```

The default font sizes for the above 3270 screen models are:

Standard 3270 Screen Model	Default Font Size
Model 2	16px
Model 3	14px
Model 4	10px
Model 5	12px

Modifying the Font Type

As a rule, you should only use monospace fonts such as Courier New or Lucida Console. With these fonts, all characters have the same width. Otherwise, when using variable-width fonts, the output will appear deformed.

If you want to define a different font type, you should define the same font type for the body, the output fields and the input fields as shown in the following example:

```
body {
  background-color: #F3F5F0;
  font-family: Lucida Console;
}

.OutputField {
  white-space:pre;
  border-width:0;
  font-family: Lucida Console;
  font-size: 100%;
}

.InputField {
  background-color: white;
  font-family: Lucida Console;
  border-width: 1px;
  font-size: 100%;
  border-color: #A7A9AB;
}
```

Defining Underlined and Blinking Text

The following elements are available:

Element Name	Description
.natTextDecoUnderline	Defines underlined text.
.natTextDecoBlinking	Defines blinking text.
.natTextDecoNormal	Defines normal text (no underline, no blinking).

Example:

```
/* Text decoration */  
.natTextDecoUnderline { text-decoration:underline; }  
.natTextDecoBlinking {text-decoration:blink; }  
.natTextDecoNormal {text-decoration:normal;}
```

Blinking text is not supported by the Internet Explorer.

Defining Italic Text

The following elements are available:

Element Name	Description
.natFontStyleItalic	Defines italic text.
.natFontStyleNormal	Defines normal text (no italics).

Example:

```
/* font style */  
.natFontStyleItalic {font-style:italic;}  
.natFontStyleNormal {font-style:normal;}
```

Defining Bold Text

The following elements are available:

Element Name	Description
.natFontWeightBold	Defines bold text.
.natFontWeightNormal	Defines normal text (not bold).

```
/* Font weight */  
.natFontWeightBold {font-weight:bolder;}  
.natFontWeightNormal {font-weight:normal;}
```

When you define bold text (`{font-weight:bolder;}`) for the default font Courier New, your text always has the same width as with normal text (`{font-weight:normal;}`).

However, when you define bold text for Courier or Lucida Console, the bold text will be wider than the normal text and your output may thus appear deformed. It is therefore recommended that you switch off bold text for Courier and Lucida Console:

```
.natFontWeightBold {font-weight:normal;}
```

Defining Different Styles for Output Fields

The following elements are available:

Element Name	Description
.FieldVariableBased	Defines the style for output fields that are based on a variable.
.FieldLiteralBased	Defines the style for output fields that are based on a literal.

Example:

```
.FieldVariableBased {
  /* font-style:italic; */
}

.FieldLiteralBased {
  /* font-style:normal; */
}
```



Note: In the above example, as well as in the standard CSS files delivered by Software AG, the variable-based output fields are defined as italic, but are commented out.

Modifying the Natural Windows

The following elements are available:

Element Name	Description
.naturalwindow	Controls the rendering of the Natural windows.
.wintitle	Controls the rendering of the titles of the Natural windows.

Example:

```
.naturalwindow {
  border-style: solid;
  border-width: 1px;
  border-color: white;
  background-color: black;
}

.wintitle {
  left: 0px;
```

```
top: 1px;
height: 17px;
width: 100%;
color: black;
font-size: 100%;
font-weight: bold;
background-color: white;
text-align: center;
font-family: Verdana;
border-bottom-style: solid;
border-bottom-width: 2px;
}
```

 **Note:** In a mainframe environment, you have to set the Natural profile parameter `WEBIO` accordingly to enable this feature. See *WEBIO - Web I/O Interface Screen Rendering in the Parameter Reference* which is provided with Natural for Mainframes.

Modifying the Message Line

The rendering of the message line is controlled by the `.MessageLine` element.

Example:

```
.MessageLine {
  color: blue;
}
```

 **Note:** In a mainframe environment, you have to set the Natural profile parameter `WEBIO` accordingly to enable this feature. See *WEBIO - Web I/O Interface Screen Rendering in the Parameter Reference* which is provided with Natural for Mainframes.

Modifying the Background Color

The background color is defined in the `body` element.

Example:

```
body {  
  background-color: #F3F5F0;  
  font-family: Lucida Console;  
}
```

Modifying the Color Attributes

You can define different colors for all Natural color attributes. These are:

- Red
- Green
- Blue
- Yellow
- White
- Black
- Pink
- Turquoise
- Transparent

You can define these color attributes for input fields and output fields, and for normal output and reverse video.

The following examples show how to define the color attribute “Red”.

Define the color for a normal output field:

```
.natOutputRed {color: darkred;}
```

Define the foreground and background colors for an output field with reverse video:

```
.reverseOutputRed {background-color: darkred; color:#F3F5F0;}
```

Define the color for a normal input field:

```
.natInputRed {color: darkred;}
```

Define the foreground and background colors for an input field with reverse video:

```
.reverseInputRed {background-color: darkred; color:#F3F5F0;}
```

Modifying the Style of the PF Key Buttons

The following elements are available:

Element Name	Description
.PFButton	Controls the style for normal rendering.
.PFButton:hover	Controls the style that is used when the mouse hovers over a PF key button.

Example:

```
.PFButton {
  text-align: center;
  width: 90px;
  border-style: ridge;
  border-width: 3px;
  padding: 2px;
  text-decoration: none;
  font-family: Verdana;
  font-size: 12px;
  height: 22px;
}

.PFButton:hover {
  color: #FFFF00;
  background-color: #222222;
}
```



Note: In a mainframe environment, you have to set the Natural profile parameter `WEBIO` accordingly to enable this feature. See *WEBIO - Web I/O Interface Screen Rendering in the Parameter Reference* which is provided with Natural for Mainframes.

XSLT Files

In addition to the CSS files described above, the Natural Web I/O Interface client uses XSLT files with specific names for the conversion of the Natural Web I/O Interface screens from the internal XML format to HTML. The following elements are affected:

- Input text is placed into the HTML element `<input>`.
- Output text is placed into the HTML element `<input>` (with attribute `readonly="readonly"`).
- A message line is placed into the HTML element ``.

- PF keys are embedded in an XML island and then rendered with JavaScript.
- Window elements are embedded in an XML island and then rendered with JavaScript.



Note: The JavaScript file which is part of the above conversion is *natunicript.js*. It is located in the *scripts* directory which can be found in the `<installdir>/natuniweb` directory.

The name of the default XSLT file is:

- *transuni.xsl* for all supported browsers.

The default XSLT file can be found in the following directory:

`<installdir>/natuniweb/web-INF`

The XSLT file is only read once when the server is started.



Important: It is recommended that you do not change the above XSLT file. Software AG may change or correct the original XSLT transformations in new versions or service packs of the product.

You can copy your own XSLT file into the above directory. In this case, the file must have the following name:

- *usertransuni.xsl* for all supported browsers.

If this user file can be found when the server is started, it is read instead of the default XSLT file.

When you make changes to this file, you have to restart the server so that your changes become effective.

10 Starting a Natural Application with a URL

The connection parameters available in the configuration file for the session and on the logon page can also be specified as URL parameters of the logon page URL. This allows bookmarking the startup URL of a Natural application or starting an application by clicking a hyperlink in a document.

The URL parameters overrule the definitions in the configuration file, with the exception described in the table below.

The following URL parameters are available for the logon page:

URL Parameter	Corresponding Option in the Session Configuration
<code>natsession</code>	Session ID
<code>natserver</code>	Host name
<code>natport</code>	Port number
<code>natuser</code>	User name
<code>natprog</code>	Application
<code>natparam</code>	Natural parameters
<code>natparamext</code>	Natural parameters The URL parameter <code>natparamext</code> extends an existing Natural parameter definition in the configuration file. The extension works in the following way: the Natural parameters defined in the configuration file come first. Then, the Natural parameters defined in the URL parameter <code>natparamext</code> are added, separated by a space character. If you want to overrule the definition in the configuration file, use the URL parameter <code>natparam</code> instead.
<code>nattimeout</code>	Timeout (n seconds)



Important: All parameter values must be URL-encoded.

Example: In order to start the Natural program `dump`, while your application server is running on `myhost:8080` and your Natural Web I/O Interface server is running on `myserver1:4811`, you can use the following URL:

`http://myhost:8080/natuniweb/natural.jsp?natserver=myserver1&natport=4811&natprog=dump&natuser=my-username`

11

Configuring Container-Managed Security

▪ General Information	76
▪ Name and Location of the Configuration File	76
▪ Activating Security	76
▪ Defining Security Constraints	77
▪ Defining Roles	77
▪ Selecting the Authentication Method	78
▪ Configuring the UserDatabaseRealm	78

General Information

The Natural Web I/O Interface client comes as a Java EE-based application. For the ease of installation, the access to this application is by default not secured. You might, however, wish to restrict the access to certain parts of the application to certain users. An important example is the **configuration tool**, which enables you to modify the Natural session definitions and the logging configuration of the Natural Web I/O Interface client. Another example is the Natural logon page.

This section does not cover the concepts of JAAS-based security in full extent. It provides, however, sufficient information to activate the preconfigured security settings of the Natural Web I/O Interface client and to adapt them to your requirements.

Name and Location of the Configuration File

Security is configured in the file *web.xml*. This file is located in the following directory:

```
<tomcat-install-dir>/webapps/natuniweb/WEB-INF
```

Activating Security

Great care must be taken when editing and changing the configuration file *web.xml*. After a change, the application server must be restarted.

Edit the file *web.xml* and look for the section that is commented with "Uncomment the next lines to add security constraints and roles.". Uncomment this section by removing the comment marks shown in boldface below:

```
<!-- Uncomment the next lines to add security constraints and roles. -->
<!--
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Configuration Tool</web-resource-name>
    <url-pattern>/conf_index.jsp</url-pattern>
    <url-pattern>/faces/*</url-pattern>
  </web-resource-collection>
  ...
<security-role>
  <description>Administrator</description>
  <role-name>nwoadmin</role-name>
</security-role>
-->
```

Defining Security Constraints

The security constraints defined by default are just examples. A `<security-constraint>` element contains a number of `<web-resource-collection>` elements combined with an `<auth-constraint>` element. The `<auth-constraint>` element contains a `<role-name>`. The whole `<security-constraint>` element describes which roles have access to the specified resources.

Example - the following definition specifies that only users in the role "nwoadmin" have access to the configuration tool:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Configuration Tool</web-resource-name>
    <url-pattern>/conf_index.jsp</url-pattern>
    <url-pattern>/faces/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>nwoadmin</role-name>
  </auth-constraint>
</security-constraint>
```

In the following section, you will see where and how the roles are defined.

Defining Roles

A few lines below in the file *web.xml*, there is a section `<security-role>`. Here, the roles that can be used in `<security-constraint>` elements are defined. You can define additional roles as needed. The assignment of users to roles is done outside this file and will often be done in a user management that is already established at your site.

Example:

```
<security-role>
  <description>Administrator</description>
  <role-name>nwoadmin</role-name>
</security-role>
```

Selecting the Authentication Method

In the file *web.xml*, there is a section `<login-config>`. The only element that should possibly be adapted here is `<auth-method>`. You can choose between the authentication methods "FORM" and "BASIC". Form-based authentication displays a specific page on which users who try to access a restricted resource can authenticate themselves. Basic authentication advises the web browser to retrieve the user credentials with its own dialog box.

Example:

```
<login-config>
  <auth-method>FORM</auth-method>
  ...
</login-config>
```

Configuring the UserDatabaseRealm

In the *tomcat-users.xml* file (which is located in the *conf* directory), specify the role "nwoadmin" for any desired user name and password. For example:

```
<user username="pepe" password="pepe123" roles="nwoadmin"/>
```

For detailed information on the necessary realm configuration for Tomcat, see <http://tomcat.apache.org/tomcat-6.0-doc/realm-howto.html#UserDatabaseRealm>.

12

Configuring SSL

- General Information 80
- Creating Your Own Trust File 80
- Defining SSL Usage in the Configuration File 81

General Information

Trust files are used for a secure connection between the Natural Web I/O Interface server and the Natural Web I/O Interface client. Server authentication cannot be switched off. A trust file is always required.

A trust file contains the certificates that you trust. These can be certificates of a CA (Certificate Authority) such as VeriSign, or self-signed certificates.

For information on the steps that are required on the Natural Web I/O Interface server and how to generate a self-signed certificate which needs to be imported to the client, see [SSL Support](#).

To establish a secure connection, you have to proceed as described in the topics below.

Creating Your Own Trust File

To create your own trust file, you can use, for example, Sun's keytool utility which can be found in the *bin* directory of the Java Runtime Environment (JRE). Here are some helpful examples:

- Create an empty, password-protected trust file:

```
keytool -genkey -alias foo -keystore truststore.jks -storepass "your-password"  
keytool -delete -alias foo -keystore truststore.jks
```

- Import a certificate:

```
keytool -import -alias "name-for-ca" -keystore truststore.jks -storepass ←  
"your-password" -file server.cert.crt
```

You should use a meaningful name for the alias.

- List the certificates in a trust file:

```
keytool -list -v -keystore truststore.jks
```

- Delete a certificate from a trust file:

```
keytool -delete -alias "name-for-ca" -keystore truststore.jks
```

When you modify the trust file or its password, you have to restart the application server so that your modification takes effect.

Defining SSL Usage in the Configuration File

Invoke the **configuration tool** and proceed as follows:

1. In the global settings for all defined sessions, define the **SSL trust file path** and, if required, the **SSL trust file password**. See also *Global Settings* in *Natural Client Configuration Tool*.

With the server authentication, the Natural Web I/O Interface client checks whether the certificate of the Natural Web I/O Interface server is known. If it is not known, the connection is rejected.

When a trust file is not defined in the configuration tool, the Natural Web I/O Interface client tries to read the file *calist* from the *lib/security* directory of the Java Runtime Environment (JRE). The default password for this file is "changeit".

2. Define a session and set the session option **Use SSL** to **Yes**. See also *Overview of Session Options* in *Natural Client Configuration Tool*.

13 Logging

- General Information 84
- Name and Location of the Configuration File 84
- Invoking the Logging Configuration Page 84
- Overview of Options for the Output File 86

General Information

The Natural Web I/O Interface client uses the Java Logging API. In case of problems with the Natural Web I/O Interface client, you can enable logging and thus write the logging information to an output file. This should only be done when requested by Software AG support.

You configure logging using the [configuration tool](#).



Note: Some logging information is also written to the console, regardless of the settings in the configuration file. The console shows the information which is normally provided by the logging levels SEVERE, WARNING and INFO.

Name and Location of the Configuration File

The name of the configuration file is *natlogger.xml*, which is located in:

`<application-server-install-dir>server/default/deploy/naturalunicode.rar/log`

Invoking the Logging Configuration Page

The content of the configuration file *natlogger.xml* is managed using the **Logging Configuration** page of the [configuration tool](#).

➤ **To invoke the Logging Configuration page**

- 1 In the frame on the left, choose the **Logging Configuration** link.

The **Logging Configuration** page appears in the right frame. Example:

Logging Configuration

Specify the output log file characteristics.

- "/" : The local pathname separator
- "%t": The system temporary directory
- "%h": The value of the "user.home" system property
- "%g": The generation number to distinguish rotated logs
- "%u": A unique number to resolve conflicts
- "%%": Translates to a single percent sign "%"

File pattern name:

File type:

File size (in Kbytes; 0=unlimited):

Number of files:

File enabled: Yes No

Append mode: Yes No

Specify log levels for individual modules. The available settings are:

- SEVERE: Events that interfere with normal program execution
- WARNING: Warnings, including exceptions
- INFO: Messages related to server configuration or server status, excluding errors
- CONFIG: Messages related to server configuration
- FINE: Minimal verbosity
- FINER: Moderate verbosity
- FINEST: Maximum verbosity

Communication:

Resource adapter:

Session beans:

Message beans:

Configuration file:

Logging:

Utilities:

Natural Web I/O Interface pages:

- Specify the characteristics of the output file as described below in the section [Overview of Options for the Output File](#).
- Specify the log levels for individual modules by selecting the log level from the corresponding drop-down list box.

A brief description for each log level is provided on the **Logging Configuration** page.

- Choose the **Save Configuration** button to write the modifications to the configuration file.



Caution: When you do not choose the **Save Configuration** button but log out instead or leave the configuration tool by entering another URL, your modifications are not written to the configuration file.

Overview of Options for the Output File

The following options are provided for specifying the characteristics of the output file:

Option	Description
File pattern name	<p>The pattern for generating the output file name. Default: "%h/nwolog%g.log".</p> <p>The default value means that an output file with the name <i>nwolog<number>.log</i> will be created in the home directory of the user who has started the application server.</p> <p>For detailed information on how to specify the pattern, see the Java API documentation.</p>
File type	<p>The format of the output file. Select one of the following entries from the drop-down list box:</p> <ul style="list-style-type: none"> ■ Text format Output in simple text format (default). ■ XML format Output in XML format. <p>The corresponding formatter class is then used.</p>
File size	<p>The maximum number of bytes that is to be written to an output file. Zero (0) means that there is no limit. Default: "0".</p>
Number of files	<p>The number of output files to be used. This value must be at least "1". Default: "10".</p>
File enabled	<p>If set to Yes (default), the file handler is enabled. If set to No, the file handler is disabled.</p>
Append mode	<p>If set to Yes, the logging information is appended to the existing output file. If set to No (default), the logging information is written to a new output file.</p>