Natural

Natural Web I/O Interface

Version 6.3.12 for OpenVMS

October 2012

This document applies to Natural Version 6.3.12 for OpenVMS.

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

Copyright © 1984-2012 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their licensors.

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

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://documentation.softwareag.com/legal/ 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 and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". This document is part of the product documentation, located at http://documentation.softwareag.com/legal/ and/or in the root installation directory of the licensed product(s).

Document ID: NATOV-NNATWEBIO-6312-20121005

Table of Contents

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 an OpenVMS environment.
Installing the Natural Web I/O Interface Client	How to install the Natural Web I/O Interface client on a web/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 an OpenVMS environment. For information on how to install it in a mainframe, UNIX or Windows environment, see the Natural documentation for the appropriate platform.

I Introduction

1 Introduction

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.

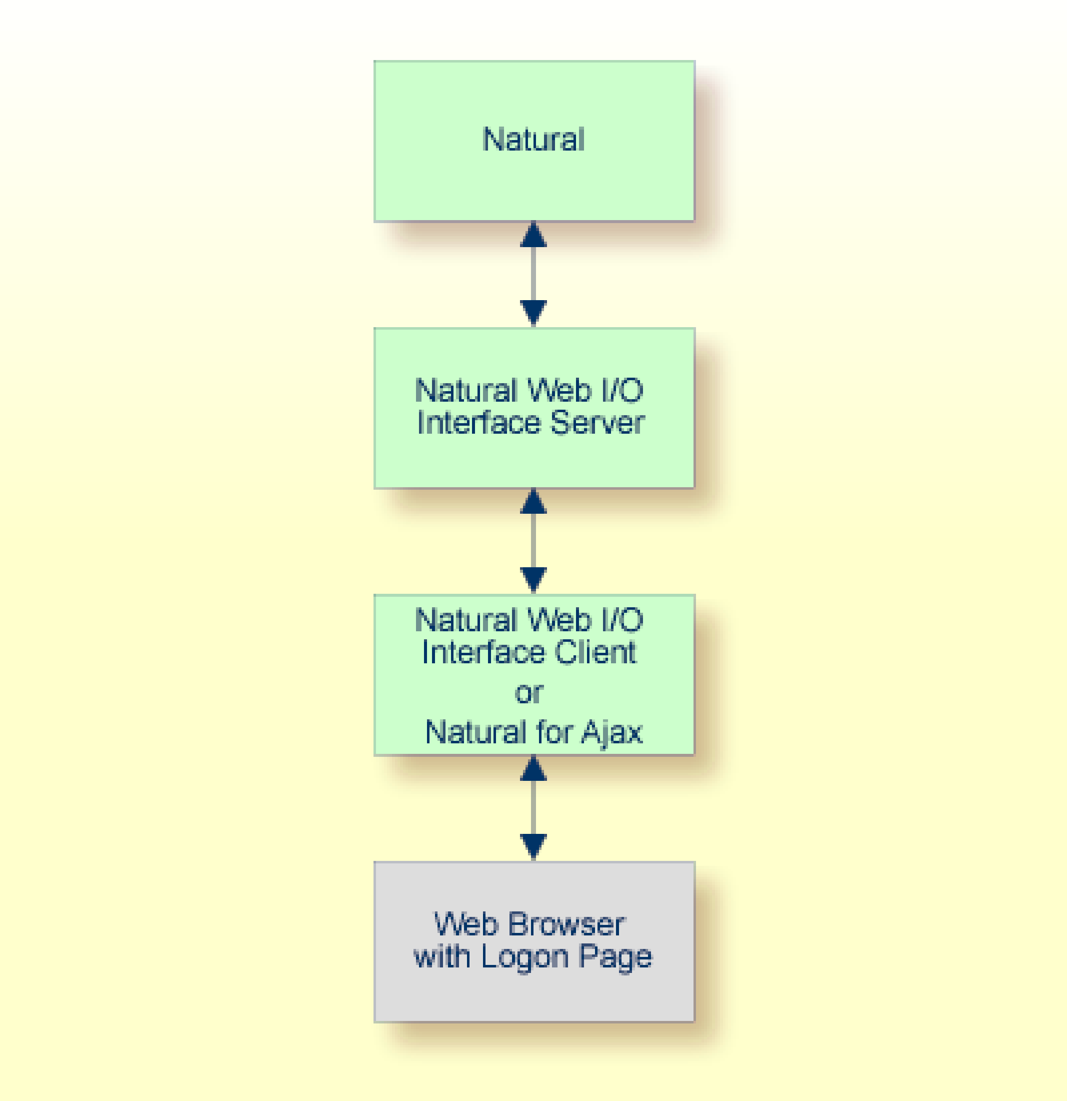
Two types of client are supported:

- 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:


Natural
Natural Web I/O Interface Server
Natural Web I/O Interface Client or Natural for Ajax
Web Browser with Logon Page

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 techiques 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.
- Using the mouse pointer, it is possible to position the cursor on an output-only field, however, the returned cursor position (system variable `*CURSOR`) is always on the first field character.
- 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 only. Other cursor movements with the other arrow keys (for example, within output fields or to the next input field or vertical movements) are not possible.
- 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. The reverse attribute is currently only supported in the implementation for Microsoft's Internet Information Services (IIS).
- 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.

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.
- The program attention keys (PA1, PA2 and PA3) are simulated by the key combinations CTRL+F1, CTRL+F2, CTRL+F3.
- The clear key is simulated by CTRL+F4,

- 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.
- 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.

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

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

On OpenVMS, the server part of the Natural Web I/O Interface is implemented as a daemon.

Installing the Natural Web I/O Interface Daemon

The installation of the Natural Web I/O Interface daemon is part of the Natural for OpenVMS installation.

Directories

The following directories are created when Natural is installed together with the Natural Web I/O Interface on an OpenVMS system:

Directory	Description
SAG$ROOT:[NATURAL]	Top-level Natural directory (*NATDIR*).
NATDIR:[V`<version>`]	Directory with all components for the current Natural version.
NATDIR:[V`<version>`.INSTALL]	Contains the command procedures for the post-installation.
NATDIR:[V`<version>`.NWO]	Contains the command procedure *START_NWOSRVD.COM* for starting the daemon.
NATDIR:[NWO.`<nodename>`]	Work directory. Contains the template file *NWOSRVD.CONF*, the generated command procedure, and the log file for each daemon.
NWODIR:[NWOEXUEX.USEREXIT1]	Contains the files for building the *LIBNWOUSEREXIT1.EXE*.
NWODIR:[NWOEXUEX.USEREXIT2]	Contains the files for building the *LIBNWOUSEREXIT2.EXE*.

Note: The above table lists the most important directories and files.

Environment Variables	Description
NATDIR="SAG$ROOT:[NATURAL]"	Top-level Natural directory.
NATVERS	Current Natural version.
NWODIR="NATDIR:[NWO]"	Top-level Natural Web I/O Interface directory.
NWONODE	Name of the machine.

Configuring the Natural Web I/O Interface Daemon on OpenVMS

When the Natural installation has finished, the directory *NATDIR:[NWO.*`<nodename>`*]* contains the file *NWOSRVD.CONF*.

The configuration of the Natural Web I/O Interface daemon can be done using the Natural Web I/O Interface daemon commands or by editing the configuration file *NWOSRVD.CONF*.

The following topics are covered below:

- Natural Web I/O Interface Daemon Commands
- NWOSRVD.CONF - Configuration File for the Natural Web I/O Interface Daemon
- START_NWOSRVD.COM - Command Procedure for Starting the Natural Web I/O Interface Daemon
- Environment Variables

Natural Web I/O Interface Daemon Commands

The following commands can be specified at the OpenVMS command prompt:

Command	Description
`nwosrvd -help`	Shows all available Natural Web I/O Interface daemon commands and subcommands.
`nwosrvd -v`	Shows the version of the Natural Web I/O Interface daemon.
`nwosrvd` *`nnnn`*	Defines the listening port number.
`nwosrvd -show`	Shows the configuration of the Natural Web I/O Interface daemon.

`nwosrvd -config` *`keys`*	Changes the configuration of the Natural Web I/O Interface daemon. The following keys can be specified:
	`-userexit1=`*`pathname`*
	The message defined with this key is saved in the `UserExit1` key of the configuration file *NWOSRVD.CONF*, section `[UserExits]`.
	`-userexit2=`*`pathname`*
	The message defined with this key is saved in the `UserExit2` key of the configuration file *NWOSRVD.CONF*, section `[UserExits]`.
	`-logging=`*`option`*
	The option defined with this key is saved in the `Logging` key of the configuration file *nwosrvd.conf*, section `[Miscellaneous]`.
	`-ssl=[yes|no]`
	The option defined with this key is saved in the `ssl` key of the configuration file *nwosrvd.conf*, section `[SSL]`.
	To remove any user exits from the configuration, enter the following command:
	`nwosrvd -config -userexit1=`

Command	Description
	Once the configuration was changed, the Natural Web I/O Interface daemon must be restarted.

NWOSRVD.CONF - Configuration File for the Natural Web I/O Interface Daemon

The configuration file *NWOSRVD.CONF* contains information that the user exits need for the Natural Web I/O Interface daemon. It has the following content:

```
[Miscellaneous]
Logging=I

[UserExits]
;UserExit1=DISK$USER:[DIR.NWOEXUEX]LIBNWOUSEREXIT1.EXE
;UserExit2=DISK$USER:[DIR.NWOEXUEX]LIBNWOUSEREXIT2.EXE

[SSL]
ssl=no
```

Section in Configuration File	Description
`[Miscellaneous]`	The key `Logging` is used to define the amount of logging information that is to be reported. One of the following options can be specified: `E` for errors. `W` for warnings. `I` for information. See also *Logging Information*.

`[UserExits]`	Two user exits can be defined:
	`UserExit1`
	The library that is defined by `UserExit1` contains the following function:
	`int nwo_CheckUsernameAndPassword(const char *pUsername, const char *pPassword, const char *pNewPassword, char *pErrorMessage)`
	If the key `UserExit1` is defined in the configuration file, the function `nwo_CheckUsernameAndPassword` is responsible for checking the user name and password. If a new password is received, user exit 1 is also responsible for changing the password.
	In the case of an error, the return code of the function must be "0"; in this case, the `pErrorMessage` is returned to the client.
	When user name and password are correct, the return code must be a value other than "0".

Section in Configuration File	Description
	UserExit2 The library that is defined by UserExit2 contains the following functions: ■ int nwo_Messages(int *iNumberOfMessages, char *pMessage[]) iNumberOfMessages: Number of messages returned in the array. pMessage: Array of messages. If the key UserExit2 is defined in the configuration file, the function nwo_Messages is called when a new connection (client) is accepted and the messages returned by this function are sent to the client. User exit 2 may be used, for example, to send a message such as the following: “For maintenance reasons, the Natural application XXXXX will be down next monday, from 18:00 until 19:00”. In the case of an error, the return code of the function must be "0". After the function nwo_Messages has been called, the function nwo_FreeMessages is called. ■ int nwo_FreeMessages(int iNumberOfMessages, char *pMessage[]) iNumberOfMessages: Number of messages. pMessage: Array of messages. If the key UserExit2 is defined, the function nwo_FreeMessages is called to free any resources (normally memory) allocated in the function nwo_Messages. In the case of an error, the return code of the function must be "0".
[SSL]	The key ssl is used to define whether the SSL protocol is to be used. One of the following values can be specified: "yes" or "no". See also *SSL Support*.

START_NWOSRVD.COM - Command Procedure for Starting the Natural Web I/O Interface Daemon

The command procedure *START_NWOSRVD.COM* is used to start the Natural Web I/O Interface daemon. For further information, see ***Starting and Stopping the Natural Web I/O Interface Daemon***.

Environment Variables

In the command procedure *START_NWOSRVD.COM*, several environment variables can be set for the Natural session that is started by the daemon:

DEFINE 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.

DEFINE 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 daemon command**:

```
nwosrvd -config -logging=option
```

`option` can be one of the following:

Option	Description
`E`	Error. When this option is specified, the Natural Web I/O Interface daemon reports only errors. In the case of an error, the daemon usually exits immediately.
`W`	Warning. When this option is specified, the Natural Web I/O Interface daemon reports errors and warnings for uncritical errors. In the case of a warning, the daemon continues to run.
`I`	Information.

Option	Description
	When this option is specified, the Natural Web I/O Interface daemon reports errors, warnings and information. The information messages allow to check the session parameters, IP address, etc.

Help information, for example, on how to run, configure and install the Natural Web I/O Interface daemon is always provided. The messages which inform you when the daemon 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 daemon command:

```
nwosrvd -show
```

The logging messages are shown directly for the standard output. The format of the messages is as in the following example:

```
%NWOSRVD-E: 18.01.2008 14:55:20 NWO_SRVDCONF is not established.
```

The following information is provided:

- `%NWOSRVD` is the internal name of the Natural Web I/O Interface daemon.
- The message type is shown directly after `%NWOSRVD`. It can be one of the following: `-E` (error), `-W` (warning), `-I` (information), or `-H` (help).
- Date and time when the message was reported.
- Any text or message which pertains to the error, warning, information or help.

If you want to save these messages, you have to redirect the standard output to a file.

Example:

```
DEFINE/USER SYS$OUTPUT file-name
nwosrvd -show
```

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

- Generate a certificate signing request and a private RSA key:

```
@SSL$COM:SSL$CERT_TOOL
```

In the resulting menu, choose **3. Create a Certificate Signing Request** and follow the instructions on the screen.

- Generate a self-signed certificate:

```
@SSL$COM:SSL$CERT_TOOL
```

In the resulting menu, choose **4. Create a Self-Signed Certificate** and follow the instructions on the screen.

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

- Put the generated files from *SSL$KEY* into *NATDIR:[NWO.`<nodename>`]*. Change the default file names so that they contain "cert.": *server.cert.csr*, *server.cert.crt* and *server.cert.key*.
- Import the generated files to a truststore on the client. See also ***Trust Files (Java EE-based servers only)***.

Working with the OpenVMS Components of the Natural Web I/O Interface

The OpenVMS components of the Natural Web I/O Interface are used to start the Natural applications linked with the Natural Web I/O Interface library.

The following topics are covered below:

- Starting and Stopping the Natural Web I/O Interface Daemon
- Starting a Natural Application

Starting and Stopping the Natural Web I/O Interface Daemon

The Natural Web I/O Interface daemons are responsible for accepting new sessions.

Before you start the Natural Web I/O Interface daemon, you have to make sure that all requirements as described in the section *General Information for Starting the Daemons* of the *Installation* documentation are met.

The command procedure *START_NWOSRVD.COM* is used to start the Natural Web I/O Interface daemon:

```
@NATDIR:[V<version>.NWO]START_NWOSRVD.COM <portnumber>
```

The process which is created by the command procedure *START_NWOSRVD.COM* has the name *NWOSRVD_*`<portnumber>` (for example, *NWOSRVD_2800*).

Note: The daemon must be started on a port which is not yet used.

To stop the daemon process, enter the following command:

```
STOP NWOSRVD_<portnumber>
```

Starting a Natural Application

See *Configuring the Client* for information on how to set up the client.

III Installing the Natural Web I/O Interface Client

This part explains how to install the Natural Web I/O Interface client on an application server or web server 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:

3 Prerequisites

Application Server or Web Server

The following application/web servers are supported. The application/web servers are not delivered with the Natural Web I/O Interface.

- **Java EE-based Server**
  The following application servers are supported.
  - **Sun Java System Application Server 9.1**
    This can be downloaded from ***http://www.oracle.com/technetwork/index.html***.

    On this application server, the Natural Web I/O Interface client consists of an enterprise application (*natuniapp.ear*) and a resource adapter (*naturalunicode.rar*). Both components are installed on the application server.

    Note: Sun Java Application Server 9.1 is also known as Oracle GlassFish 2.

  - **Oracle GlassFish Server 3.1**
    This can be downloaded from ***http://glassfish.java.net/public/downloadsindex.html***.

    On this application server, the Natural Web I/O Interface client consists of a web application (*natuniweb*). This component is installed on the application server.

    Oracle GlassFish Server supports the following profiles: "Full Platform Profile" and "Web Profile". Both profiles can be used with the Natural Web I/O Interface client.

  - **JBoss Application Server 4.2**
    This can be downloaded from ***http://www.jboss.org/***.

    On this application server, the Natural Web I/O Interface client consists of an enterprise application (*natuniapp.ear*) and a resource adapter (*naturalunicode.rar*). Both components are installed on the application server.

  The prerequisite for using the Natural Web I/O Interface client on these application servers is that Java 6 Update 24 or above is installed. When you install on Oracle GlassFish Server, specific minimum versions are required, depending on your GlassFish version.

- **Microsoft Internet Information Services (IIS)**
  On this web server, the Natural Web I/O Interface client is based on the ASP.NET 2.0 technology. Therefore, the prerequisite for running the Natural Web I/O Interface client with IIS is that the .NET framework 2.0 is installed on the same server on which Microsoft Internet Information Services is installed.

  When ASP.NET 2.0 has not yet been installed, you can download the .NET Framework Version 2.0 from:

http://www.microsoft.com/download/en/details.aspx?id=19

You have to install the Natural Web I/O Interface client on one of these servers as described later in this documentation.

Note: Server farms are not supported.

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 6 (see *http://tomcat.apache.org/*).

Apache Ant

Apache Ant 1.8.1 or above is required to perform the deployment on JBoss Application Server. This tool is freely available on *http://ant.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 4.2.3 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 6.2.5 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 6.3.3 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 7 through 9.
- Mozilla Firefox 3.6 through 10.

  Note: Mozilla Firefox 10 (Extended Support Release) is supported. In future versions, only the Extended Support Releases of Mozilla Firefox will be explicitly supported.

- Safari 5.1 on Windows and Mac OS X.

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

4 Installing the Natural Web I/O Interface Client on Sun Java System Application Server

If you want to use the Natural Web I/O Interface client with Sun Java System Application Server, you must proceed as described in this chapter.

Installation Steps

The Natural Web I/O Interface client is installed using the Administration Console of Sun Java System Application Server.

The following is assumed:

- `<host>` is the name of the machine on which the application server is installed.
- `<port>` is the name of the port where the application server is installed. In a default installation, this is port 8080.
- `<adminport>` is the name of the port where the Administration Console is installed. In a default installation, this is port 4848.
- `<sunas>` is the path to the directory in which the application server is installed. In a default installation on Windows, this is *C:\Sun\AppServer*.

The following topics are covered below:

- First-time Installation
- Update Installation

First-time Installation

To install the Natural Web I/O Interface client

1 Download the Natural Web I/O Interface client for Sun Java System Application Server from Empower (***https://empower.softwareag.com/***) and unzip the contents to a directory of your choice on your hard disk.

Or:

Natural for UNIX, Natural for OpenVMS and Natural for Windows: Copy the complete contents of the *nwo/*`<platform>`*/j2ee/v*`<nnnn>`*/sun-apps* directory from the installation medium 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 Edit the file `<sunas>`*/domains/domain1/config/server.policy* and add the following setting in order to enable the Java Logging API:

```
grant {
 permission java.util.logging.LoggingPermission "control";
};
```

⚠ **Important:** If you do not add this setting, the resource adapter will not start and the Natural Web I/O Interface client will therefore be inoperative.

4 Make sure that the application server is running.

5 Open your web browser and enter the following URL:

```
http://<host>:<adminport>
```

This opens the Administration Console.

6 Deploy the resource adapter *naturalunicode.rar*:

1. Open the tree node **Applications > Connector Modules**.
2. Choose **Deploy**.
3. Select *naturalunicode.rar* as the package file to be uploaded to the application server.
4. Choose **Next**. "naturalunicode" is automatically included as the application name.
5. Choose **Finish**.

7 Define the JNDI name for the resource adapter:

1. Open the tree node **Resources > Connectors > Connector Connection Pools**.
2. Choose **New**.
3. Enter "NatPool" (the name is arbitrary) as the name.
4. Select **naturalunicode** as the resource adapter.
5. Each connection to a Natural host results in a new connection being made. Since each user requires a unique host session, connection pooling cannot be used. Therefore, you should make sure there are enough sessions for your users. The default maximum number is "32".
6. Choose **Next**.
7. Choose **Next**.
8. Choose **Finish**.
9. Open the tree node **Resources > Connectors > Connector Resources**.
10. Choose **New**.
11. Enter "eis/NaturalUnicodeRA" as the JNDI name.
12. Select **NatPool** (or whatever name you specified previously) as the pool name.
13. Choose **OK**.

8 Deploy the enterprise application *natuniapp.ear*:

1. Open the tree node **Applications > Enterprise Applications**.
2. Choose **Deploy**.
3. Select *natuniapp.ear* as the file to upload.
4. Choose **Next**.
5. Choose **OK**. The deployment may take several minutes.

9 Restart the application server.

Update Installation

To update the Natural Web I/O Interface client

1 Make a backup copy of your *sessions.xml* file which is located in *../AppServer/domains/domain1/applications/j2ee-apps/natuniapp/natuniweb_war/WEB-INF*. If you have changed any other files (such as style sheets), also make backup copies of these files.

2 Download the Natural Web I/O Interface client for Sun Java System Application Server from Empower (***https://empower.softwareag.com/***) and unzip the contents to a directory of your choice on your hard disk.

Or:

Natural for UNIX, Natural for OpenVMS and Natural for Windows: Copy the complete contents of the *nwo/*`<platform>`*/j2ee/v*`<nnnn>`*/sun-apps* directory from the installation medium to a directory of your choice on your hard disk.

3 On UNIX platforms: Dearchive the TAR file using the following command:

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

4 Edit the file `<sunas>`*/domains/domain1/config/server.policy* and make sure that the following setting has been defined:

```
grant {
 permission java.util.logging.LoggingPermission "control";
};
```

⚠ **Important:** This setting is required as of Natural Web I/O Interface Version 1.3.2. If this setting is missing, the resource adapter will not start and the Natural Web I/O Interface client will therefore be inoperative.

5 Make sure that the application server is running.

6 Open your web browser and enter the following URL:

```
http://<host>:<adminport>
```

This opens the Administration Console.

7 Undeploy the resource adapter *naturalunicode.rar*:

1. Open the tree node **Resources > Connectors > Connector Connection Pools**.
2. Mark the check box **Natpool** (or the check box for whatever name you specified previously).
3. Choose **Delete**.
4. Open the tree node **Applications > Connector Modules**.
5. Mark the check box **naturalunicode**.
6. Choose **Undeploy**.

8 Undeploy the enterprise application *natuniapp.ear*:

1. Open the tree node **Applications > Enterprise Applications**.
2. Mark the check box **natuniapp**.
3. Choose **Undeploy**.

9 Deploy the resource adapter *naturalunicode.rar* as in a first-time installation.

10 Define the JNDI name for the resource adapter as in a first-time installation.

11 Deploy the enterprise application *natuniapp.ear* as in a first-time installation.

12 Copy your backup files back to the required places.

13 Restart the application server.

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 Installing the Natural Web I/O Interface Client on Oracle GlassFish Server

Note: If you want to install on Oracle GlassFish 2.0, see ***Installing the Natural Web I/O Interface Client on Sun Java System Application Server***. The information below applies to Oracle GlassFish 3.1.

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

Installation Steps

Although Oracle GlassFish Server is the successor of Sun Application Server, the installation is different from the one described for Sun Application Server. On Oracle GlassFish Server, the Natural Web I/O Interface client consists of a web application (*natuniweb.war*).

The Natural Web I/O Interface client is installed using the Administration Console of Oracle GlassFish Server. It can also be installed automatically (autodeploy).

The following is assumed:

- `<host>` is the name of the machine on which the application server is installed.
- `<port>` is the name of the port where the application server is installed. In a default installation, this is port 8080.
- `<adminport>` is the name of the port where the Administration Console is installed. In a default installation, this is port 4848.
- `<glassfish>` is the path to the directory in which the application server is installed. In a default installation on Windows, this is *C:\glassfish3\glassfish*.

The following topics are covered below:

- First-time Installation
- Update Installation

First-time Installation

To install the Natural Web I/O Interface client using the Administration Console

1 Download the Natural Web I/O Interface client for Oracle GlassFish Server from Empower (***https://empower.softwareag.com/***) and unzip the contents to a directory of your choice on your hard disk.

Or:

Natural for UNIX, Natural for OpenVMS and Natural for Windows: Copy the complete contents of the *nwo/*`<platform>`*/j2ee/v*`<nnnn>`*/GlassFish* directory from the installation medium 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 the application server is running.

4 Open your web browser and enter the following URL:

```
http://<host>:<adminport>
```

This opens the Administration Console.

5 Deploy the web application:

1. Select the tree node **Applications**.
2. Choose **Browse**.
3. Go to the directory containing the web application *natuniweb.war* and select the web application.
4. Choose **OK**.

To install the Natural Web I/O Interface client automatically

1 Download the Natural Web I/O Interface client for Oracle GlassFish Server from Empower (*https://empower.softwareag.com/*) and unzip the contents to a directory of your choice on your hard disk.

Or:

Natural for UNIX, Natural for OpenVMS and Natural for Windows: Copy the complete contents of the *nwo/*`<platform>`*/j2ee/v*`<nnnn>`*/GlassFish* directory from the installation medium 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 Start the application server.

4 Copy the web application *natuniweb.war* into the directory `<glassfish>`*\domains\domain1\autodeploy*.

Update Installation

To update the Natural Web I/O Interface client

1. Make a backup copy of your *sessions.xml* file which is located in *<glassfish>/domains/domain1/applications/natuniweb/WEB-INF/lib*. If you have changed any other files (such as style sheets), also make backup copies of these files.
2. Download the Natural Web I/O Interface client for Oracle GlassFish Server from Empower (*https://empower.softwareag.com/*) and unzip the contents to a directory of your choice on your hard disk.

   Or:

   Natural for UNIX, Natural for OpenVMS and Natural for Windows: Copy the complete contents of the *nwo/<platform>/j2ee/v<nnnn>/GlassFish* directory from the installation medium to a directory of your choice on your hard disk.
3. On UNIX platforms: Dearchive the TAR file using the following command:

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

4. Make sure that the application server is running.
5. Open your web browser and enter the following URL:

   ```
   http://<host>:<adminport>
   ```

   This opens the Administration Console.
6. Select the tree node **Applications** and undeploy the natuniweb application.
7. Deploy the new version of the web application *natuniweb.war* as in a first-time installation.

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***.

6 Installing the Natural Web I/O Interface Client on JBoss Application Server

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

Installation Steps

Only one version of the Natural Web I/O Interface client can be installed on the same JBoss Application Server.

You can either install the Natural Web I/O Interface client or Natural for Ajax on the same JBoss Application Server, not both.

It is assumed that `<jboss>` is the directory of your JBoss Application Server installation.

The following topics are covered below:

- First-time Installation
- Update Installation

First-time Installation

To install the Natural Web I/O Interface client

1 Download the Natural Web I/O Interface client for JBoss Application Server from Empower (***https://empower.softwareag.com/***) and unzip the contents to a directory of your choice on your hard disk.

Or:

Natural for UNIX, Natural for OpenVMS and Natural for Windows: Copy the complete contents of the *nwo/*`<platform>`*/j2ee/v*`<nnnn>`*/jboss* directory from the installation medium 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 Install Apache Ant (you need Apache Ant to deploy the Natural Web I/O Interface client to the JBoss Application Server; see the ***Prerequisites*** above for the required version number):

1. Download and unzip Apache Ant (from ***http://ant.apache.org/***) into an installation directory of your choice. Avoid a directory name that contains blanks.
2. Let the environment variable `ANT_HOME` point to the directory `<ant>` (where `<ant>` is the directory of your Ant installation).
3. Add `<ant>`*/bin* to your `PATH` environment variable.

4 Deploy the Natural Web I/O Interface client to JBoss Application Server:

1. Copy the the Natural Web I/O Interface client distributables to a directory on a disk drive.
2. In the directory that contains the the Natural Web I/O Interface client distributables, there is an Ant script named *jbossdeploynwo.xml*. Edit this script and change the setting

   ```
   <property name="jbosshome" value="C:/Program Files/Jboss/jboss-4.2.2.GA"/>
   ```

   to

   ```
   <property name="jbosshome" value="<jboss>"/>
   ```

   where `<jboss>` is your JBoss Application Server installation directory.

   Important: Take care to use forward slashes (also on Windows) when specifying the directory path.

3. Execute the script *jbossdeploynwo.xml* by entering the following command:

   ```
   ant -f jbossdeploynwo.xml
   ```

   Wait for the message “BUILD SUCCESSFUL”. This indicates that the deployment was successful.

5 Edit the file `<jboss>`*/server/default/deploy/jbossjca-service.xml* and change the setting

```
<!-- Enable connection close debug monitoring -->
<attribute name="Debug">true</attribute>
```

to

```
<!-- Enable connection close debug monitoring -->
<attribute name="Debug">false</attribute>
```

6 Start JBoss Application Server.

Update Installation

To update the Natural Web I/O Interface client

1 Download the Natural Web I/O Interface client for JBoss Application Server from Empower (*https://empower.softwareag.com/*) and unzip the contents to a directory of your choice on your hard disk.

Or:

Natural for UNIX, Natural for OpenVMS and Natural for Windows: Copy the complete contents of the *nwo/<platform>/j2ee/v<nnnn>/jboss* directory from the installation medium 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 JBoss Application Server.
4. Deploy the Natural Web I/O Interface client to JBoss Application Server as in a first-time installation.
5. Make sure that the file *<jboss>/server/default/deploy/jbossjca-service.xml* contains the same settings as described for a first-time installation.
6. Start JBoss Application Server.

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***.

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

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:

- `<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 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.

Or:

Natural for UNIX, Natural for OpenVMS and Natural for Windows: Copy the complete contents of the *nwo/*`<platform>`*/j2ee/v*`<nnnn>`*/tomcat* directory from the installation medium 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 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.

Or:

Natural for UNIX, Natural for OpenVMS and Natural for Windows: Copy the complete contents of the *nwo/*`<platform>`*/j2ee/v*`<nnnn>`*/tomcat* directory from the installation medium 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***.

8 Installing the Natural Web I/O Interface Client on Microsoft Internet Information Services (IIS)

If you want to use the Natural Web I/O Interface client with Microsoft Internet Information Services (IIS), you must proceed as described in this chapter.

Installation Steps

The descriptions below apply when working with Windows XP. The procedure may be slightly different when working with a different Windows version.

The following topics are covered below:

- First-time Installation
- Update Installation

First-time Installation

To install the Natural Web I/O Interface client

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

Or:

Natural for UNIX, Natural for OpenVMS and Natural for Windows: Copy the complete contents of the *nwo/windows/iis/v<nnnn>* directory from the installation medium to a directory of your choice on your hard disk.

2 Open the Control Panel and choose **Administrative Tools > Internet Information Services**.

3 In the resulting window, expand the node with the name of your computer in the tree and select the node **Web Sites > Default Web Site**.

4 Open the context menu and choose **New > Virtual Directory**.

Or:

From the **Action** menu, choose **New > Virtual Directory**.

A wizard for creating a new virtual directory appears.

5 Choose the **Next** button to proceed to the next page of the wizard.

6 Enter an alias name for the virtual directory (for example, "Natural") and choose the **Next** button.

⚠ **Important:** If you want to use the Natural Execution and Debugging plug-in for Eclipse, the alias name for the virtual directory must be "Natural".

7 Enter the path to the directory which contains the files you have copied to your hard disk and choose the **Next** button.

8 Leave the access permissions with the default values (that is: **Read** and **Run script** are selected) and choose the **Next** button.

9 Choose the **Finish** button.

The virtual directory is now created on the IIS server.

10 Select the name of the new virtual directory in the tree.

11 Open the context menu and choose **Properties**.

Or:

From the **Action** menu, choose **New > Properties**.

12 In the resulting dialog box, choose the **ASP.NET** page. Select the ASP.NET version. It has to be at least 2.0.50727 (the release version of ASP.NET 2.0). Choose the **OK** button to close the dialog box.

Update Installation

To update the Natural Web I/O Interface client

1 On your hard disk, create a new directory for the update version that you want to install.

2 Download the Natural Web I/O Interface client for IIS from Empower (***https://empower.softwareag.com/***) and unzip the contents to the new directory.

Or:

Natural for UNIX, Natural for OpenVMS and Natural for Windows: Copy the complete contents of the *nwo/windows/iis/v*<nnnn> directory from the installation medium to a directory of your choice on your hard disk.

3 Open the Control Panel and choose **Administrative Tools > Internet Information Services**.

4 Open the properties for the existing virtual directory (for example, "Natural"). On the **Virtual Directory** page, change the path so that it points to the new directory which contains the update version.

5 Copy your *settings.xml file* from the directory which contains the previous version to the new directory (the *settings.xml* file is located in the root of your installation directory). If you have changed any other files (such as style sheets), also copy these files to the new directory.

Installation Verification

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

To verify the installation

- When you have defined the alias name "Natural" in the IIS (see above), you can start the Natural Web I/O Interface client in the browser using the following URL:

  Enter the following URL in your web browser:

  ```
  http://<host>:<port>/Natural/Default.aspx
  ```

  For example:

  ```
  http://myhost:8080/Natural/Default.aspx
  ```

  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***.

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

With a future version of the Natural Web I/O Interface, Microsoft Internet Information Services (IIS) will no longer be supported. If you are currently using the Natural Web I/O Interface client on IIS, you will have to move to another supported server platform. This may be JBoss Application Server, Oracle GlassFish Server or Apache Tomcat.

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 is 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. With a later version of the Natural Web I/O Interface, the chapter ***Overview of Configuration File Elements*** will be removed from the documentation.

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/realm-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:

10 About the Logon Page

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 *Managing the Configuration File for the Session*).

In order to start a Natural application from the logon page, you enter one of the URLs listed below inside your browser. Different names and locations are used for the logon pages, depending on the application/web server that you are using.

- **Java EE-based Server**
  You have to enter the following URL:

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

- **Microsoft Internet Information Services (IIS)**
  You have to enter the following URL:

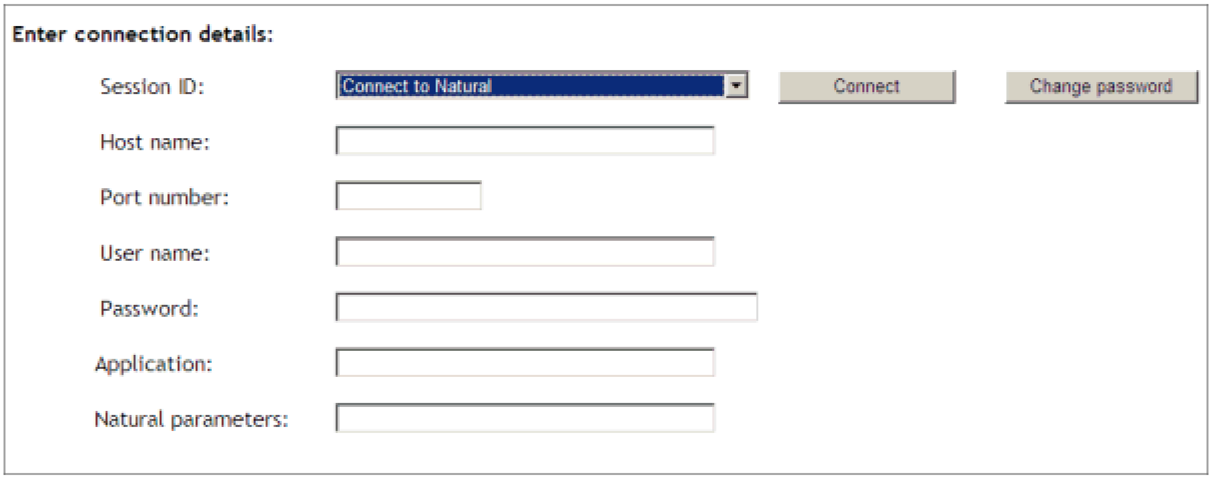
  ```
  http://<host>:<port>/Natural/Default.aspx
  ```

where `<host>` and `<port>` are the host name and port number of your application/web 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.

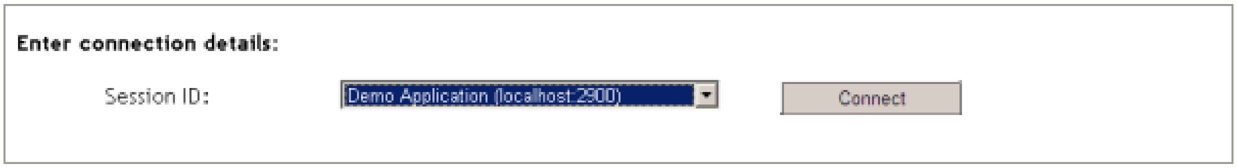
The following example shows part of a logon page which results from a configuration file in which no special entries are defined for a session:



Enter connection details:

Session ID: Connect to Natural

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: Demo Application (localhost:2900)

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** (Java EE-based servers only) and in the **configuration file**, 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.

11 Managing the Configuration File for the Session

General Information

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

The session configurations can be created in the following ways:

- When the Natural Web I/O Interface client has been installed on a Java EE-based server, 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.
- You can edit the XML file manually. See ***Overview of Configuration File Elements***.

  When the Natural Web I/O Interface client has been installed on IIS, this is the only way to change the XML file.

Name and Location of the Configuration File

Different names and locations are used for the configuration files, depending on the application/web server that you are using.

- **Java EE-based Server**
  The name of the configuration file is *sessions.xml*. It can be found in the *WEB-INF* directory. The path to this directory depends on the application server that you are using.
  - **JBoss Application Server**
    `<application-server-install-dir>`*/server/default/deploy/natuniapp.ear/natuniweb.war/WEB-INF*
  - **Sun Java System Application Server**
    `<application-server-install-dir>`*/domains/domain1/applications/j2ee-apps/natuniapp/natuniweb_war/WEB-INF*
  - **Oracle GlassFish Server**
    `<application-server-install-dir>`*/glassfish/domains/domain1/applications/natuniweb/WEB-INF*
  - **Apache Tomcat**
    `<application-server-install-dir>`*/webapps/natuniweb/WEB-INF*
- **Microsoft Internet Information Services (IIS)**
  The name of the configuration file is *settings.xml*. It can be found in the root directory of your application in IIS.

12 Using the Configuration Tool (Java EE-based servers only)

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 on a Java EE-based server.

To invoke the configuration tool

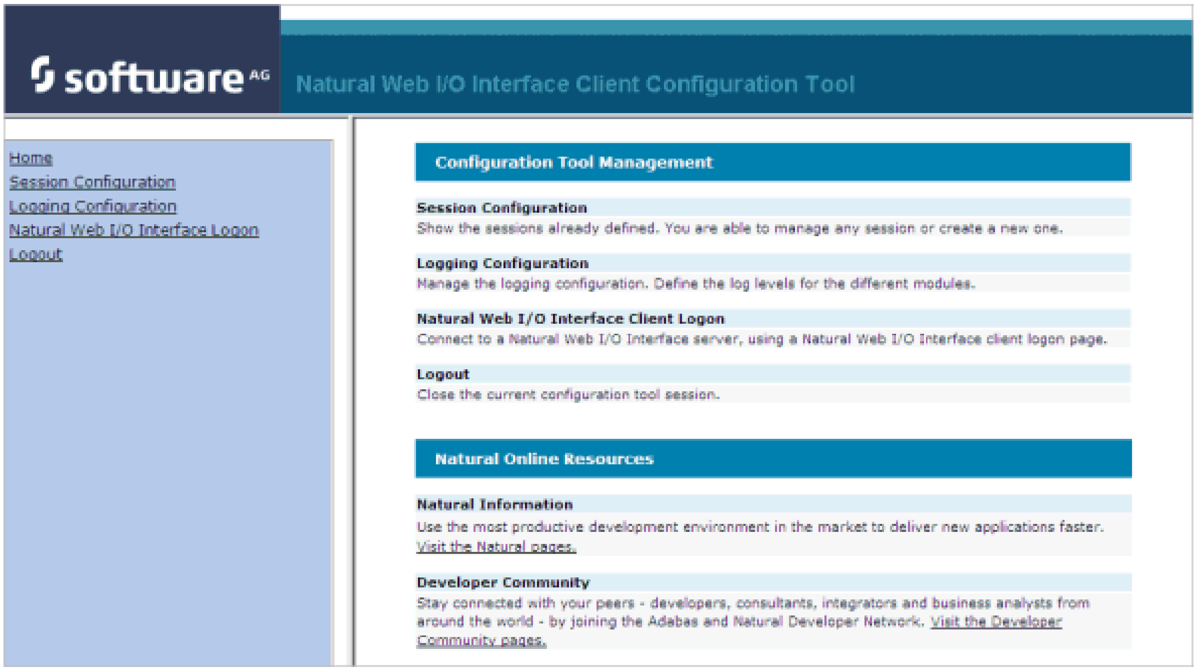
- Enter the following URL in your browser:

  ```
  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 (Java EE-based servers only)*** 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.



software AG
Natural Web I/O Interface Client Configuration Tool
Home
Session Configuration
Logging Configuration
Natural Web I/O Interface Logon
Logout

Configuration Tool Management

Session Configuration
Show the sessions already defined. You are able to manage any session or create a new one.

Logging Configuration
Manage the logging configuration. Define the log levels for the different modules.

Natural Web I/O Interface Client Logon
Connect to a Natural Web I/O Interface server, using a Natural Web I/O Interface client logon page.

Logout
Close the current configuration tool session.

Natural Online Resources

Natural Information
Use the most productive development environment in the market to deliver new applications faster. Visit the Natural pages.

Developer Community
Stay connected with your peers - developers, consultants, integrators and business analysts from around the world - by joining the Adabas and Natural Developer Network. Visit the Developer Community pages.

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:

- Saving the Configuration

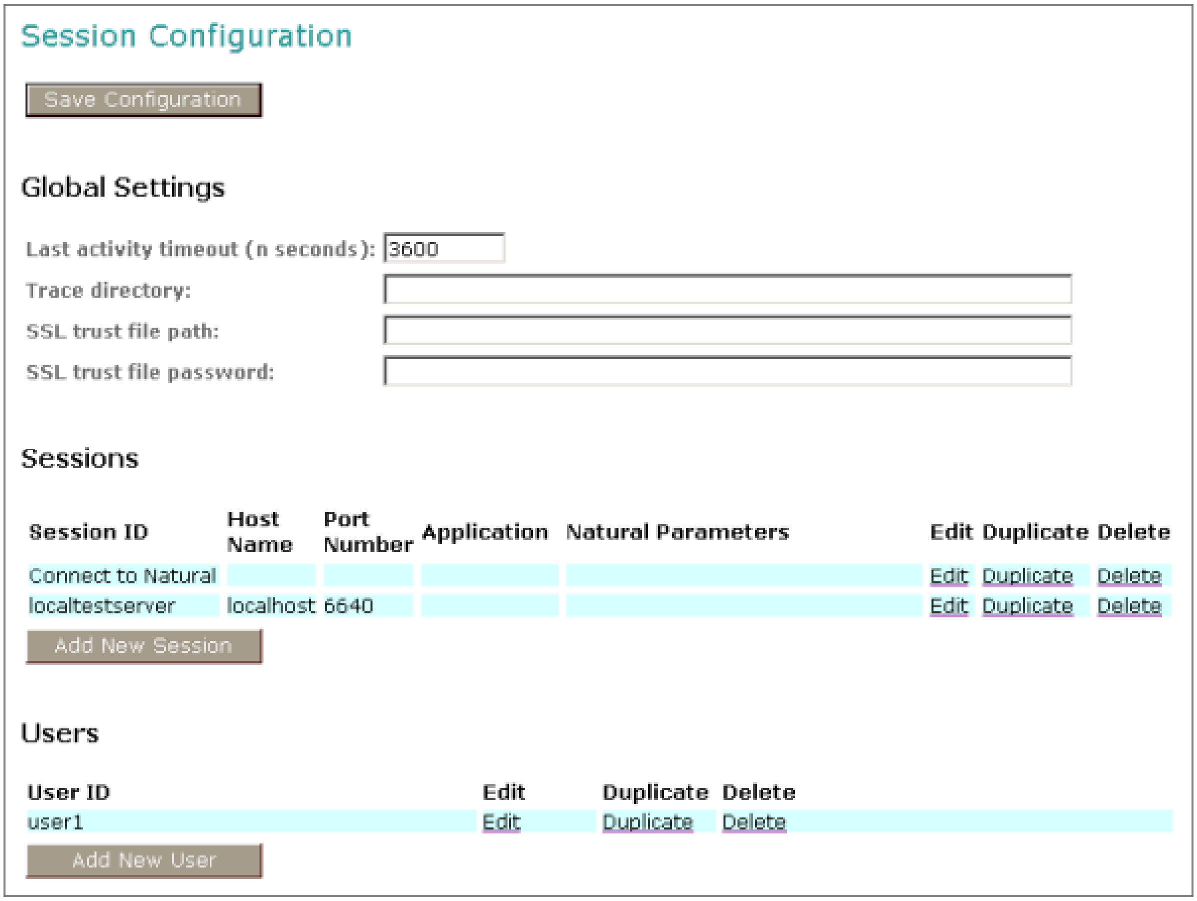
Invoking the Session Configuration Page

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): 3600
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)	Timeout after the last activity of the user in seconds. The default is 3600 seconds (1 hour). When the defined number of seconds has been reached, the session is closed. You can also set an individual timeout value for each session (see *Overview of Session Options* below).

Trace directory	Optional. Location of a different trace directory.
	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 `java.io.tmpdir`. On Windows, this is normally the environment variable `TMP` for the user who started the application server. On UNIX, this is normally */tmp* or */var/tmp*.
	You can also set this property in the start script for the application server. The following examples apply to JBoss.
	- Example for Windows (*run.bat*):
	`set JAVA_OPTS=%JAVA_OPTS% -Djava.io.tmpdir=C:\temp`
	- Example for UNIX (*run.sh*):
	`set JAVA_OPTS="$JAVA_OPTS -Djava.io.tmpdir=/tmp`
	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 *Trust Files (Java EE-based servers only)* 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.

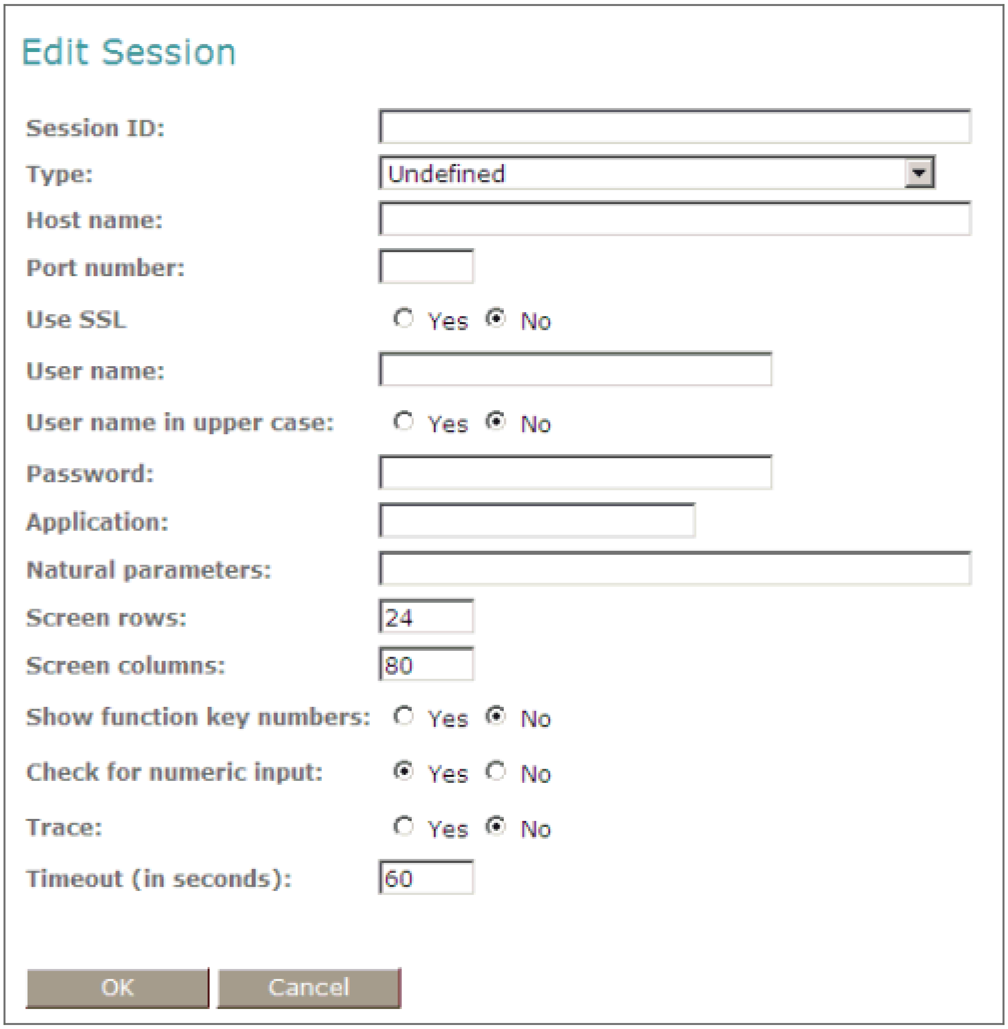
4 Choose the **Save Configuration** button to write the modifications to the configuration file.

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:	
Type:	Undefined
Host name:	
Port number:	
Use SSL	○ Yes ● No
User name:	
User name in upper case:	○ Yes ● No
Password:	
Application:	
Natural parameters:	
Screen rows:	24
Screen columns:	80
Show function key numbers:	○ Yes ● No
Check for numeric input:	● Yes ○ No
Trace:	○ Yes ● No
Timeout (in seconds):	60

Note: If you need to set the protocol version for the Natural Web I/O Interface (which is only required for older Natural versions), you have to do this manually in the configuration file. See the description of the `protocol` attribute in the section ***Overview of Configuration File Elements***.

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.

Option	Description
Type	The platform on which user ID and password are authenticated. You can select the required setting from the drop-down list box. ■ **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 "*domain\user-ID*").
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	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. **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.
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	Optional. A valid password for the above user ID. Under Windows, UNIX and OpenVMS, this is always the operating system password of the user, even if Natural Security is active. 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. When a password is specified, the corresponding field does not appear on the logon page. The configuration tool saves the password in encrypted form.

Option	Description
Application	■ **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: `TEST01 data1,data2` ■ **Natural for UNIX** The name of the UNIX shell script for starting the Natural application (a file similar to *nwo.sh*). ■ **Natural for OpenVMS** The name of the Natural image file (for example, *natural<version>* or *natural<version>.exe*). ■ **Natural for Windows** The name of the Windows command file (*.bat*) for starting the Natural application. When this is specified, the corresponding field does not appear on the logon page.

Natural parameters	Optional. Parameters for starting the Natural application. This can be stack parameters, a parameter file/module or other Natural-specific information.
	■ **Natural for Mainframes**
	Used to pass dynamic Natural profile parameters to the session, for example:
	`SYSPARM=(MYPARMS) STACK=(LOGON MYAPPL)`
	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 `STACK`.
	■ **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:
	`PARM=MYPARM STACK=(LOGON MYLIB;MENU)`
	■ **Natural for OpenVMS**
	Used for starting a Natural application, for example:
	`BP=BP`*`node-name`*` NLDCHK WEBIO=ON "STACK=(LOGON SYSEXT;MENU)"`

Screen rows	The number of rows in the output window. Possible values: minimum 24, no upper limit. Default: 24. Not used by Natural for Mainframes which uses the profile parameter `TMODEL` instead.
Screen columns	The number of columns in the output window. Possible values: minimum 80, no upper limit. Default: 80. Not used by Natural for Mainframes which uses the profile parameter `TMODEL` instead.
Show function key numbers	If set to **Yes**, the PF key numbers are shown next to the PF keys.
Trace	Should only be set to **Yes** when requested by Software AG support.

Option	Description
Check for numeric input	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"): `blank` + (plus) - (minus) _ (underscore) , (comma . (period) ? (question mark) If set to **No**, numeric input fields are not validated.
Timeout (n seconds)	The number of seconds that the client waits for an answer from Natural after an update of a page was sent to Natural. The default is 60 seconds. Normally, you need not change this default value.

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 passwort

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 (Java EE-based servers only)* 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.

13 Overview of Configuration File Elements

Contents of the Configuration File

By default, the configuration file is delivered with the following entries:

```
<?xml version="1.0" encoding="utf-8" ?>
<settings>
     <global>
        <last_activity_timeout>3600</last_activity_timeout>
     </global>
    <sessions>
        <session id="Connect to Natural" trace="false">
        </session>
        <session id="localtestserver" trace="false">
            <natural_server>localhost</natural_server>
            <natural_port>6640</natural_port>
            <natural_program/>
            <user id="" pwd=""/>
            <natural_parameter/>
        </session>
    </sessions>
</settings>
```

Note: With Natural for Mainframes, it is recommended to specify the Natural program that starts the application in the element `natural_program` instead of passing it with the profile parameter `STACK`.

The first session that is defined in the above configuration file has the name "Connect to Natural". This session is automatically preselected in the logon page. Since no further elements are defined for this session, all required input fields for logging on to a session are shown on the logon page and the user has to specify all required information.

Note: The `global` section containing the timeout value is only available in the configuration file for a Java EE-based server. See also ***Setting the Timeout***.

To add a new session definition to the configuration file, you must add a new `session` element. It should have the following minimum entries:

```
<session id="your-session-name" trace="false">
  <natural_server>your-Natural-server</natural_server>
  <natural_port>port-ID-of-Natural-server</natural_port>
  <natural_program>Natural-program-name</natural_program>
</session>
```

Sessions

The following table explains the elements and attributes that can be used in the `sessions` section of the configuration file for adding a new session.

Element Name	Attribute Name	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.
`session`	`type`	Optional. The platform on which user ID and password are authenticated. Possible values: ■ **MF** Natural for Mainframes. User ID and password can have a maximum of 8 characters. ■ **MF-NSC** 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. ■ **OS** Natural for UNIX, Natural for OpenVMS or Natural for Windows: 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 "*domain\user-ID*"). Default: "OS".
`session`	`protocol`	Optional. The current protocol version for the Natural Web I/O Interface is 7. The latest Natural versions automatically use the appropriate protocol version. For older Natural versions, it is required to define the appropriate protocol version in the configuration file: ■ Natural versions 6.2.1 through 6.2.4 (UNIX) require protocol version 2. As of version 6.2.5 it is no longer required to define a protocol version. ■ Natural versions 6.3.1 and 6.3.2 (UNIX) require protocol version 3. As of version 6.3.3, it is no longer required to define a protocol version. ■ Natural versions up to version 4.2.3.2 (mainframe) require protocol version 3.

Element Name	Attribute Name	Description
		As of version 4.2.3.3 and 4.2.4, it is no longer required to define a protocol version. Possible values: 2, 3, 4, 5, 6, and 7. Default: 7.
`session`	`ssl`	Java EE-based servers only. Optional. If set to "true", 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. **Important:** If you want to use SSL with Natural for Mainframes, the type "MF" or "MF-NSC" must be defined; the type must not be "OS". The other way around, if you want to use SSL with Natural for Windows, UNIX or OpenVMS, the type must be "OS". Default: "false".
`session`	`trace`	Optional. Should only be set to "true" when requested by Software AG support. For further information, see *Tracing*. Default: "false".
`user`	`id`	Optional. A valid user ID for the current machine. When this attribute is specified, the corresponding field does not appear on the logon page.
`user`	`pwd`	Optional. A valid password for the above user ID. When this attribute is specified, the corresponding field does not appear on the logon page.
`user`	`encrypted`	Java EE-based servers only. Used by the configuration tool which stores the password in encrypted form.
`user`	`ucase`	Optional. If set to "true", the input field for the user ID is in upper-case mode. Default: "false".
`natural_server`		The name or TCP/IP address of the server on which Natural and the Natural Web I/O Interface daemon (Natural for UNIX and Natural for OpenVMS) or the Natural Web I/O Interface server (Natural for Mainframes) or the Natural Web I/O Interface service (Natural for Windows) are running. When this element is specified, the corresponding field does not appear on the logon page.
`natural_port`		The TCP/IP port number on which the Natural Web I/O Interface daemon (Natural for UNIX and Natural for OpenVMS) or the Natural Web I/O Interface server (Natural for Mainframes) or the Natural Web I/O Interface service (Natural

Element Name	Attribute Name	Description
		for Windows) is listening. When this element is specified, the corresponding field does not appear on the logon page.
`natural_program`		■ **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: `TEST01 data1,data2` ■ **Natural for UNIX** The name of the UNIX shell script for starting the Natural application (a file similar to *nwo.sh*). ■ **Natural for OpenVMS** The name of the Natural image file (for example, *natural<version>* or *natural<version>.exe*). ■ **Natural for Windows** The name of the Windows command file (*.bat*) for starting the Natural application. When this element is specified, the corresponding field does not appear on the logon page.

`natural_parameter`		Optional. Parameters for starting the Natural application. This can be stack parameters, a parameter file/module or other Natural-specific information.
		■ **Natural for Mainframes**
		This element is used to pass dynamic Natural profile parameters to the session, for example:
		`SYSPARM=(MYPARMS) STACK=(LOGON MYAPPL)`
		Note: It is recommended to specify the Natural program that starts the application in the element `natural_program` instead of passing it with the profile parameter `STACK`.
		■ **Natural for UNIX and Natural for Windows**
		This element is used when the above shell script (UNIX) or command file (Windows) uses the parameter `$5` after "natural", for example:
		`PARM=MYPARM STACK=(LOGON MYLIB;MENU)`
		■ **Natural for OpenVMS**
		This element is used for starting a Natural application, for example:
		`BP=BP`*`node-name`*` NLDCHK WEBIO=ON "STACK=(LOGON SYSEXT;MENU)"`

Element Name	Attribute Name	Description
natural_parameter	visible	IIS only. If set to "true", a field containing the Natural parameters is shown on the logon page. Default: "false".
nattimeout		Java EE-based servers only. Optional. Timeout for the response from the host. See ***Setting the Timeout***.
theme		IIS only. The theme (style) that the web page is using. For a first test, you can use the theme with the name "3270Theme" (see ***Using Themes (IIS only)***). When this element is specified, the corresponding field does not appear on the logon page. When themes are shown on the logon page, they are provided in a drop-down list box.
screen	rows	Optional. The number of rows in the output window. Possible values: minimum 24, no upper limit. Default: 24. Not used by Natural for Mainframes which uses the profile parameter TMODEL instead.
screen	columns	Optional. The number of columns in the output window. Possible values: minimum 80, no upper limit. Default: 80. Not used by Natural for Mainframes which uses the profile parameter TMODEL instead.
screen	top	IIS only. The top position of the output window inside the browser. Use "0" for the very top. This value is given in Natural units, not pixels.
screen	left	IIS only. The left position of the output window inside the browser. Use "0" for the very left. This value is given in Natural units, not pixels.
screen	size	IIS only. The size of the output window. Possible values: "normal", "small", "tiny" and "extratiny". Default: "normal".
screen	pfkeypos	IIS only. The position of the PF keys. Possible values: "bottom" and "right". Default: "bottom".

Element Name	Attribute Name	Description
		Note: For a Java EE-based server, the position of the PF keys is determined in the style sheet. See ***Using Style Sheets***.
screen	showfkeynumbers	Optional. If set to "true", the PF key numbers are shown next to the PF keys. Default: "false".
screen	checknumfields	If set to "true" (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"): *blank* + (plus) - (minus) _ (underscore) , (comma . (period) ? (question mark) If set to "false", numeric input fields are not validated.

If a field is not to appear on the logon page, you can specify the corresponding element or attribute as described above. To do so, you either specify a value for the attribute or element or you omit the value. For example, you can specify either of the following:

```
<natural_program></natural_program>
```

or

```
<natural_program>sysprof</natural_program>
```

In both cases, the corresponding field does not appear on the logon page.

Only when an element name or attribute name is not mentioned at all, the corresponding field is shown on the logon page.

Global Settings

The following table explains the global settings that can be defined in the `global` section of the configuration file.

Element	Description
last_activity_timeout	The timeout after the last activity of the user. See *Setting the Timeout (Java EE-based servers only)* below.
trace_dir	Optional. Location of a different trace directory. See *Tracing* below.
trustfile_name	Java EE-based servers only. Optional. The path to your trust file. See *Trust Files (Java EE-based servers only)* for further information.
trustfile_password	Java EE-based servers only. If your trust file is password-protected, the appropriate password is required. The password can only be specified with the **configuration tool**. It is stored in encrypted form.

Setting the Timeout (Java EE-based servers only)

You can set two different timeouts:

- **Last activity timeout**
  This is the timeout after the last activity of the user. It is defined with last_activity_timeout in the global section of the configuration file and is set in seconds. The default is 3600 seconds (1 hour). When the defined number of seconds has been reached, the session is closed.

```
<?xml version="1.0" encoding="utf-8"?>
<settings>
  <global>
     <last_activity_timeout>3600</last_activity_timeout>
  </global>
  <sessions>
  ...
```

- **Natural server timeout**
  This timeout defines the number of seconds that the client waits for an answer from Natural after an update of a page was sent to Natural. It is defined in the nattimeout element of a session definition. The default is 60 seconds. Normally, you need not change this default value. This timeout can be set individually for each session.

```
...
    <session id="Test (UNIX)" trace="false">
      <natural_server>Myserver2</natural_server>
      <natural_port>4321</natural_port>
      <natural_program>test.sh</natural_program>
      <nattimeout>60</nattimeout>
    </session>
...
```

Tracing

Tracing should only be enabled when requested by Software AG support. To enable tracing, you set the `trace` attribute in a session definition to "true". Example:

```
<session id="MySession" trace="true">
```

Default Trace Directory for Java EE-based Servers

By default, the trace files are written to the directory which has been set by the Java property `java.io.tmpdir`. On Windows, this is normally the environment variable `TMP` for the user who started the application server. On UNIX, this is normally */tmp* or */var/tmp*.

You can also set this property in the start script for the application server. The following examples apply to JBoss.

- Example for Windows (*run.bat*):

  ```
  set JAVA_OPTS=%JAVA_OPTS% -Djava.io.tmpdir=C:\temp
  ```

- Example for UNIX (*run.sh*):

  ```
  set JAVA_OPTS="$JAVA_OPTS -Djava.io.tmpdir=/tmp
  ```

Default Trace Directory for IIS

By default, the trace files are written to the *Software AG\Natural WebIO\Traces* directory. The path to this directory depends on the operating system:

- **Windows XP**
  `<drive>`*:\Documents and Settings\*`<pc-name>`*\ASPNET\Local Settings\Application Data\Software AG\Natural WebIO\Traces*
- **Windows 2003 Server**
  `<drive>`*:\Documents and Settings\Default User\Application Data\Software AG\Natural WebIO\Traces*
- **Windows Vista**
  `<drive>`*:\Windows\ServiceProfiles\NetworkService\AppData\Local\Software AG\Natural WebIO\Traces*

Defining a Different Trace Directory

It is possible to define a different trace directory by defining the element `trace_dir` in the `global` element of the configuration file. Example:

```
<global>
  <trace_dir>E:\mytracedir</trace_dir>
</global>
```

When a different trace directory is not defined, the default trace directory will be used.

Users and Passwords

It is possible to predefine Natural users and their passwords. This is done in the `users` section of the configuration file.

The syntax is as follows:

```
<settings>
  ...
  <users>
    <user id="user1" pwd="password1"/>
    <user id="user2" pwd="password2"/>
    <user id="user3" pwd="password3"/>
  </users>
  ...
</settings>
```

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 (which is "password1" in the above syntax example):

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

14 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
natsession	**Session ID**
natserver	**Host name**
natport	**Port number**
natuser	**User name**
natprog	**Application**
natparam	**Natural parameters**
natparamext	**Natural parameters** The URL parameter natparamext 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 natparamext are added, separated by a space character. If you want to overrule the definition in the configuration file, use the URL parameter natparam instead.
nattimeout	**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

15 Using Style Sheets (Java EE-based servers only)

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*.

The location of the style sheet depends on the application server that you are using.

- **JBoss Application Server**
  ../natuniapp.ear/natuniweb.war/resources
- **Sun Java System Application Server**
  ../j2ee-apps/natuniapp/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
`.mainlayer`	Controls the position of the main output in the output window. Used for languages that are written from left-to-right (LTR).
`.mainlayer_rtl`	Controls the position of the main output in the output window. Used for languages that are written from right-to-left (RTL).
`.pfkeydiv`	Controls the position of the PF keys in the output window. Used for languages that are written from left-to-right (LTR).
`.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.	*model2.css*
Model 3 (80x32)	Between 31 and 40 rows.	*model3.css*
Model 4 (80x43)	41 rows or more.	*model4.css*
Model 5 (132x27)	30 rows or less, and more than 100 columns.	*model5.css*

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 `<span>`.

- 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 files which are part of the above conversion are *natunicscript-ie.js* (for Internet Explorer) and *natunicscript-ff.js* (for Firefox). They are located in the *scripts* directory which can be found in the `<installdir>`*/natuniweb* directory.

The names of the default XSLT files are:

- *transuni.xsl* for Internet Explorer.
- *transuni-ff.xsl* for Mozilla Firefox.

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

`<installdir>`*/natuniweb/web-INF*

The XSLT files are only read once when the server is started.

Important: It is recommended that you do not change the above XSLT files. 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 files into the above directory. In this case, the files must have the following names:

- *usertransuni.xsl* for Internet Explorer.
- *usertransuni-ff.xsl* for Mozilla Firefox.

When these user files can be found when the server is started, they are read instead of the default XSLT files.

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

16 Using Themes (IIS only)

General Information

The information in this chapter applies only to Microsoft Internet Information Services (IIS).

Themes are used to define the output style for the Natural application. Some themes are already delivered in the installation package.

Themes are standard functionality in ASP.NET 2.0. They must be contained in the *App_Themes* folder. For each theme, the *App_Themes* folder contains a subfolder. For example, the theme with the name "3270Thema" is contained in the folder *3270Theme*.

Each subfolder which is provided for a theme must contain the following files:

- *SkinFile.skin* (see below)
- *StyleSheet.css* (see below)

The Skin File

The skin file (*SkinFile.skin*) contains entries for Natural elements (such as fields) in the following form:

```
<asp:Label SkinId="Red" runat="server" forecolor="red"/>
<asp:Label SkinId="Green" runat="server" forecolor="green"/>
<asp:Label SkinId="Blue" runat="server" forecolor="blue" />
...
```

The table below shows the minimum settings that are required in the skin file to map the Natural screen in the Natural Web I/O Interface. This table contains the following columns:

- **Natural Attributes**
  The attributes that can be set in Natural.
- **Output Fields**
  The Natural output fields for text.
- **Input Fields**
  The Natural fields in which you can enter data.
- **Read-only Input Fields**
  When a window is displayed on a Natural screen, the input fields in the main screen are set to read-only. The input fields of the windows that are lying behind the top window are also displayed as read-only.

Natural Attributes	Output Fields	Input Fields	Read-only Input Fields
Red	`<asp:Label runat="server" SkinId="Red"/>`	`<asp:TextBox runat="server" SkinId="Red"/>`	`<asp:TextBox runat="server" SkinId="RedReadonly"/>`
Green	`<asp:Label runat="server" SkinId="Green"/>`	`<asp:TextBox runat="server" SkinId="Green"/>`	`<asp:TextBox runat="server" SkinId="GreenReadonly"/>`
Blue	`<asp:Label runat="server" SkinId="Blue"/>`	`<asp:TextBox runat="server" SkinId="Blue"/>`	`<asp:TextBox runat="server" SkinId="BlueReadonly"/>`
Pink	`<asp:Label runat="server" SkinId="Pink"/>`	`<asp:TextBox runat="server" SkinId="Pink"/>`	`<asp:TextBox runat="server" SkinId="PinkReadonly"/>`
Yellow	`<asp:Label runat="server" SkinId="Yellow"/>`	`<asp:TextBox runat="server" SkinId="Yellow"/>`	`<asp:TextBox runat="server" SkinId="YellowReadonly"/>`
Turquoise	`<asp:Label runat="server" SkinId="Turquoise"/>`	`<asp:TextBox runat="server" SkinId="Turquoise"/>`	`<asp:TextBox runat="server" SkinId="TurquoiseReadonly"/>`
White	`<asp:Label runat="server" SkinId="White"/>`	`<asp:TextBox runat="server" SkinId="White"/>`	`<asp:TextBox runat="server" SkinId="WhiteReadonly"/>`
Black	`<asp:Label runat="server" SkinId="Black"/>`	`<asp:TextBox runat="server" SkinId="Black"/>`	`<asp:TextBox runat="server" SkinId="BlackReadonly"/>`
Intensified	`<asp:Label runat="server" SkinId="Intensified"/>`	`<asp:TextBox runat="server" SkinId="Intensified"/>`	`<asp:TextBox runat="server" SkinId="IntensifiedReadonly"/>`
No attribute	`<asp:Label runat="server" SkinId="Normal"/>`	`<asp:TextBox runat="server" SkinId="Normal"/>`	`<asp:TextBox runat="server" SkinId="NormalReadonly"/>`

The information in the cells of the above table shows the settings of the ASP.NET web server controls. ASP.NET defines a lot of web server controls. The attribute `runat="server"` indicates that the web controls are interpreted at the server site.

The following controls are used in the default skin files:

Control	Description
`asp:Label`	A text output field.
`asp:TextBox`	A text input field.
`asp:Table`	A table where the field elements are the PF key buttons.
`asp:Button`	A PF key button.
`asp:Image`	An image link.
`asp:Panel`	A container for other web controls.

In addition to defining colors as shown above, the `SkinId` is also used to define the following:

Natural Element	Description	ASP.NET Skin File
Message line	Information text line in the Natural screen.	`<asp:Label runat="server" SkinId="messageline"/>`
Child window	The child windows of a Natural application.	`<asp:Panel runat="server" SkinId="WindowPanel"/>`
Child window title	The title of a Natural child window.	`<asp:TextBox runat="server" SkinId="WindowTitle"/>`
Child window shadow	The child window is shown with a shadow.	`<asp:Panel runat="server" SkinId="WinShadow"/>`
PF key table	The row of PF key buttons is defined in a table.	`<asp:Table runat="server" SkinId="PFKeys"/>`
PF key button	The definition of the PF keys.	`<asp:Button runat="server" SkinId="Pfkey"/>`
Head picture	Picture displayed at the top of the page.	`<asp:Image runat="server" SkinId="headpic"/>`

The Style Sheet

In the style sheet (*StyleSheet.css*), you can define styles according to the CSS standard. See ***http://www.w3.org/Style/CSS/***.

The following is an example style sheet which defines the background color and font of the web page:

```
body
{
background-color: transparent;
font-family: Courier New, Monospace;
}
```

Note: If you define the font size in a CSS file, this has no influence on the screen size. If the font size is changed, only smaller text is displayed.

17 Configuring Container-Managed Security (Java EE-based servers only)

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 Java EE-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. More information on the topics described in this section can be found, for instance, at ***http://www.jboss.org/jbossas/docs/*** (security on JBoss is described in the *Configuration Guide*) or ***http://download.oracle.com/javaee/1.4/tutorial/doc/index.html*** (see the chapter on security).

Name and Location of the Configuration File

Security is configured in the file *web.xml*. The path to this file depends on the application server.

- **JBoss Application Server**
  `<application-server-install-dir>`*/server/default/deploy/natuniapp.ear/natuniweb.war/WEB-INF*
- **Sun Java System Application Server**
  `<application-server-install-dir>`*/domains/domain1/applications/j2ee-apps/natuniapp/natuniweb_war/WEB-INF*
- **Oracle GlassFish Server**
  `<application-server-install-dir>`*/glassfish/domains/domain1/applications/natuniweb/WEB-INF*
- **Apache Tomcat**
  `<application-server-install-dir>`*/webapps/natuniweb/WEB-INF*

Note: The following applies for Sun Java System Application Server as of Version 9 and to Oracle GlassFish Server: After the deployment, the file *web.xml* can no longer be modified. Therefore, it is important that you unpack the WAR file before deploying it, make your changes to the *web.xml* file, repack the WAR file, and then deploy it. For this reason, unsigned WAR files are delivered for these two application servers.

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 of 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>
```

Choosing the Login Module (JBoss Application Server only)

In the directory *`<application-server-install-dir>`/server/default/conf*, there is a file named *njxnwo-login-config.xml*. The relevant part in this file is the selection of the login module specified in the `<login-module>` element and the configuration of this login module. The login module determines where the user definitions and the assignment of users to roles is maintained.

By default, the `UsersRolesLoginModule` is preconfigured. The `UsersRolesLoginModule` expects the role definitions in one file (*props/njxnwo-roles.properties*) and the user definitions (password and assignment to roles in another file (*props/njxnwo-users.properties*). An example user "admin" with the password "adminadmin" and the role "nwoadmin" is defined to begin with.

You can choose and configure a different login module, for instance, one that expects the user and role definitions in a database or in an LDAP directory, or even write a custom login module.

Defining the Security Realm and Users (Sun Java System Application Server and Oracle GlassFish Server only)

The following information applies to Sun Java System Application Server 9.1 and to Oracle GlassFish Server 3, however, the procedure is similar in other versions.

To create a new security realm and define the user

1 Sun Java System Application Server: Open the tree node **Configuration > Security > Realms**.

Oracle GlassFish Server: Open the tree node **Configuration > server-config > Security > Realms**.

2 Choose **New**.

3 Enter "NaturalWebIOAndAjaxRealm" as the name of the new realm.

4 Select `com.sun.enterprise.security.auth.realm.file.FileRealm` as the class name.

Use the following properties which are predefined for this class:

Option	Value
JAAS Context	fileRealm
Key File	${com.sun.aas.instanceRoot}/config/keyfile

5 Choose **OK**.

6 Edit the new realm `NaturalWebIOAndAjaxRealm` and choose the **Manage Users** button.

7 Choose **New**.

8 Enter the user names and the passwords for the users. The name of the group list must be "nwoadmin".

9 Choose **OK**.

Configuring the UserDatabaseRealm (Apache Tomcat only)

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***.

18 Trust Files (Java EE-based servers only)

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.

In the **configuration tool**, you define the path and, if required, the password for the trust file. 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".

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-passwort"
  keytool -delete -alias foo -keystore truststore.jks
  ```

- Import a certificate:

```
keytool -import -alias "name-for-ca" -keystore truststore.jks -storepass ↩
"your-passwort" -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.

19 Logging (Java EE-based servers only)

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*. The path to this file depends on the application server.

- **JBoss Application Server**
  `<application-server-install-dir>`*server/default/deploy/naturalunicode.rar/log*
- **Sun Java System Application Server**
  `<application-server-install-dir>`*/domains/domain1/applications/j2ee-modules/naturalunicode/log*

Logging on Sun Java System Application Server

On Sun Java System Application Server, the logging information is written to the normal server log. That is because Sun Java System Application Server uses the same Java Logging API as the Natural Web I/O Interface client. You can thus use a powerful Sun Java System Application Server tool, the Log Viewer, for analyzing the log. The Log Viewer is started from the web-based Admin Console; for further information, see the documentation of the Sun Java System Application Server.

We recommend that you disable the file handler in the configuration file *natlogger.xml*. Thus, you avoid that the logging information is written to two different log files (that is, the normal server log and the output file defined in *natlogger.xml*).

Logging on JBoss Application Server

JBoss Application Server uses a different logging API (log4j). In this case, we recommend that you enable the file handler in the configuration file *natlogger.xml*.

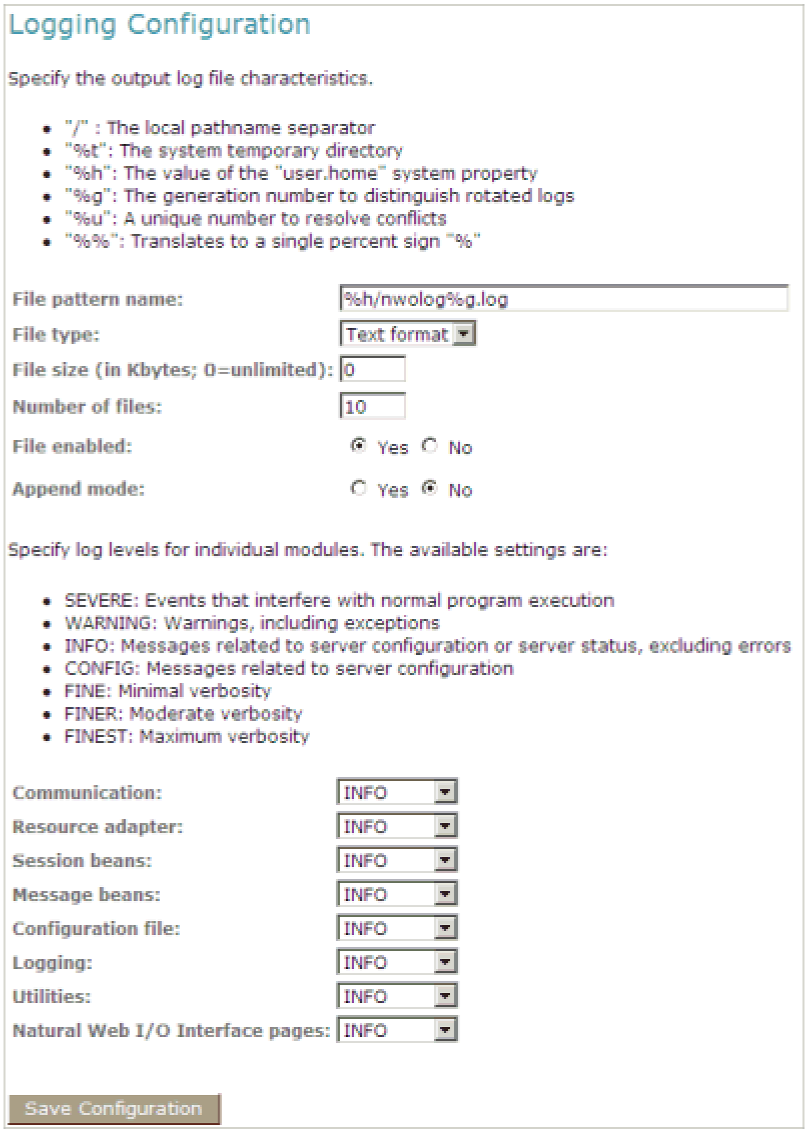
Invoking the Logging Configuration Page

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:	%h/nwolog%g.log
File type:	Text format
File size (in Kbytes; 0=unlimited):	0
Number of files:	10
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:	INFO
Resource adapter:	INFO
Session beans:	INFO
Message beans:	INFO
Configuration file:	INFO
Logging:	INFO
Utilities:	INFO
Natural Web I/O Interface pages:	INFO

2 Specify the characteristics of the output file as described below in the section *Overview of Options for the Output File*.

3 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.

4 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	The pattern for generating the output file name. Default: "%h/nwolog%g.log". The default value means that an output file with the name *nwolog*`<number>`*.log* will be created in the home directory of the user who has started the application server. For detailed information on how to specify the pattern, see the Java API documentation at ***http://download.oracle.com/javase/1.5.0/docs/api/java/util/logging/FileHandler.html***.
File type	The format of the output file. Select one of the following entries from the drop-down list box: - **Text format** Output in simple text format (default). - **XML format** Output in XML format. The corresponding formatter class is then used.
File size	The maximum number of bytes that is to be written to an output file. Zero (0) means that there is no limit. Default: "0".
Number of files	The number of output files to be used. This value must be at least "1". Default: "10".
File enabled	If set to **Yes** (default), the file handler is enabled. If set to **No**, the file handler is disabled.
Append mode	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.