Installing and Configuring the Natural Web I/O Interface Server

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

This chapter covers the following topics:

- Installing the Natural Web I/O Interface Service

- Configuring the Natural Web I/O Interface Service on Windows

- Logging Information

- SSL Support

Installing the Natural Web I/O Interface Service

The Natural Web I/O Interface service is installed with Natural for Windows if the corresponding option has been set during installation. This is part of the setup type **Development Client for Single Point of Development**. See the *Installation* documentation for further information.

Configuring the Natural Web I/O Interface Service on Windows

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

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

The following topics are covered below:

- Natural Web I/O Interface Service Commands

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

- Batch File for Starting Natural

- Environment Variables

Natural Web I/O Interface Service Commands

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

Command	Description
`nwosvcd -help`	Shows all available Natural Web I/O Interface service commands and subcommands in a message box.
`nwosvcd -install`	Installs the Natural Web I/O Interface service. The service is installed with the startup type **Automatic**. If you want to change the startup type, you must open the Windows Services management console. Under Windows XP, this is located in the Administrative Tools. Open the properties for **Software AG Natural *n.n* Web I/O Service** and select the desired startup type.
`nwosvcd -remove`	Removes the Natural Web I/O Interface service from the system.
`nwosvcd -console`	Runs the Natural Web I/O Interface service as a Windows console application. In this case, an icon appears in the notification area of the taskbar. When you invoke the context menu for this icon, the following commands are provided: ● **About** Shows the version of the Natural Web I/O Interface service. ● **Exit** Invokes a dialog box, asking whether you want to stop the Natural Web I/O Interface service or not.
`nwosvcd -show`	Shows the configuration of the Natural Web I/O Interface service.

Command	Description

`nwosvcd -config keys`	Changes the configuration of the Natural Web I/O Interface service. The following keys can be specified:
	
	-port=*nnnn*
	
	The port number.
	
	-userexit1=*pathname*
	
	The library that is defined by `userexit1` must contain the following function:
	
	`int nwo_CheckUsernameAndPassword(const char *pUsername, const char *pPassword, const char *pNewPassword, char *pErrorMessage, HANDLE *pHandleUser)`
	
	If the key `userexit1` is configured, 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".
	
	-userexit2=*pathname*
	
	The library that is defined by `userexit2` must contain the following functions:
	
	● `int nwo_Messages(int *iNumberOfMessages, char *pMessage[])`
	
	If the key `userexit2` is configured, the function `nwo_Messages` is called when a new client is accepted and the messages that are 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".
	
	-logging=*option*
	
	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*.
	
	-ssl=[yes|no]
	
	Defines whether the SSL protocol is to be used. See also *SSL Support*.
	
	To remove any user exits from the configuration, enter the following command:
	
	`nwosvcd -config -userexit1=`
	
	Once the configuration was changed, the Natural Web I/O Interface service must be restarted.

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

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

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

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

1. Open the Windows Services management console and stop the Natural Web I/O Interface service.

2. Invoke the **Command Prompt** window of Windows.

3. Go to the Natural *bin* directory which contains the file *nwosvcd.exe*.

4. Enter the following command to configure a new port number and a new user exit 1:

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

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

Batch File for Starting Natural

In order to start a Natural session, the Natural Web I/O Interface service executes a batch file. The batch file prepares the environment for the Natural session and eventually starts Natural.

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

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

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

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

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

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

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

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

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

natrt.exe %NaturalParameters% etid=$$
```

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

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

1. In the directory where *nwosvcd.exe* is located. This is the Natural *bin* directory.

2. In the system directory. This is the directory that the system function `GetSystemDirectory` would deliver.

3. In the Windows directory. This is the directory that the system function `GetWindowsDirectory` would deliver.

4. In the directories that are listed in the `PATH` environment variable.

Arguments

The batch file will receive the following arguments:

Order	Argument	Description
1	IPAddress	The client IP address from where the session is opened. **Note:** If there is a proxy, this will not be the IP address of the client workstation. Instead, it will be the IP address of the proxy.
2	ClientId	The user name from the logon page is passed as the client ID.
3	Argument3	Reserved for future use.
4	Argument4	Reserved for future use.

5	NaturalParameters	These can be any Natural parameters. The parameters are either defined in the configuration file for the session, or they are entered in the logon page. The following is an example of the corresponding entry in the configuration file:
		
		`<natural_parameter>parm=nwoparm\`
		`stack=(logon\`
		`mylib;start-program;fin)<natural_parameter>`
		
		The language that is selected in the logon page is added as the first element to the Natural parameters in the form "ulang=x".

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

Environment Variables

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

NWO_ENABLE_ACK=["YES"|"NO"]

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

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

NO   No waiting between asynchronous screens. Default value.

NWO_TIMEOUT=[*number-of-seconds*]

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

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

Logging Information

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

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

option can be one of the following:

Option	Description
E	Error. When this option is specified, the Natural Web I/O Interface service reports only errors. In the case of an error, the service usually exits immediately.
W	Warning. When this option is specified, the Natural Web I/O Interface service reports errors and warnings for uncritical errors. In the case of a warning, the service continues to run.
I	Information. When this option is specified, the Natural Web I/O Interface service 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 service is always provided. The messages which inform you when the service has been started or stopped are also part of the help information.

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

```
nwosvcd -show
```

The logging messages are saved in the Windows Event Viewer.

SSL Support

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

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

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

  Note:
  openssl.cnf and *openssl* are delivered in *%NATDIR%\%NATVERS%\bin*.

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

  ```
  set OPENSSL_CONF=%NATDIR%\%NATVERS%\bin\openssl.cnf
  ```

- Generate a certificate signing request:

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

- Generate a private RSA key:

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

- Generate a self-signed certificate:

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

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

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

- Import the generated files to a truststore on the client. See also *Trust Files (J2EE only)*.