Introduction

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

- What is the Natural Web I/O Interface?

- Components of the Natural Web I/O Interface

- Executing a Natural Application in a Web Browser

- Client-Server Compatibility

- Terminology

- Differences in a SPoD Development Environment

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

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

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. See the *Natural for Ajax* documentation.

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.

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

Executing a Natural Application in a Web Browser

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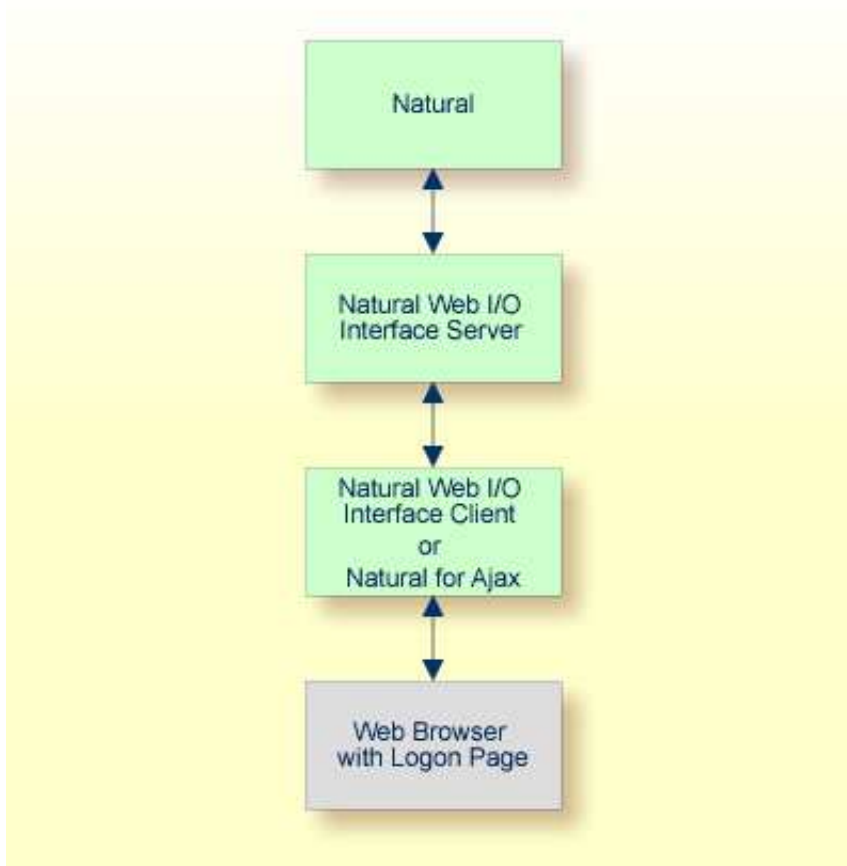
The communication steps for executing a Natural application in the web browser are:

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

   Note:
   For information on how to invoke and configure the logon page, see *Configuring the Client*.

2. The user enters all required information for starting a Natural application into the logon page. This information is sent to the client.

3. The client asks the Natural Web I/O Interface server to start the requested Natural application for this user.

4. The Natural Web I/O Interface server checks the supplied user ID and password, creates a Natural session for the user and starts the Natural application.

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

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

- **Natural Web I/O Interface Client**
  See *Browser Prerequisites* in this documentation.

- **Natural for Ajax**
  See *Browser Prerequisites* in the *Natural for Ajax* documentation.

With Firefox, you can use caret browsing. The following exception applies for Natural applications: you can only use the RIGHT-ARROW and LEFT-ARROW keys to position the cursor (also called "caret") in an output field. Caret browsing is enabled and disabled by pressing F7. For more information, see the documentation for your web browser.

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
```

- **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.

- **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.

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 43 lines and 132 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.