

Natural Web I/O Interface Server Concept and Structure

This chapter describes the concept and the structure of the server for the Natural Web I/O Interface which is designed for use on z/OS and (under SMARTS) on z/VSE, VM/CMS or BS2000/OSD.

The following topics are covered:

- Web I/O Interface Server Concept
 - Natural Web I/O Interface Server under SMARTS on BS2000/OSD
 - Front-End Stub NATRNWO
 - Front-End
 - Server Monitor
-

Web I/O Interface Server Concept

A Natural Web I/O Interface server is a multi-user, multi-tasking application. It can host Natural sessions for multiple users and execute their applications concurrently.

The concept is based on the "serverized" Natural runtime system. Its architecture comprises a server front-end stub (Web I/O Interface server stub NATRNWO) that uses the Natural front-end to dispatch Natural sessions and to execute applications within these sessions.

The Natural Web I/O Interface server architecture basically consists of:

- SMARTS runtime environment (only on z/VSE, VM/CMS and BS2000/OSD)
SMARTS is used to implement a server runtime environment for the execution of the Web I/O Interface server.
- Front-end stub
The stub NATRNWO is launched to initialize a Natural Web I/O Interface server. It listens for incoming connection requests and launches a Natural session for executing the application.
- Front-end
The front-end is called (together with the Natural runtime system) by the front-end stub for session initialization/termination, application execution and session roll-in/roll-out.
- Server monitor
A monitor task allows the administrator to control the server activities, to cancel particular user sessions or to terminate the entire server, etc.

Natural Web I/O Interface Server under SMARTS on BS2000/OSD

SMARTS is an acronym for "Software AG Multi-Architecture Runtime System". It constitutes a runtime layer that allows POSIX-like applications to run on mainframe operating systems. Software AG products communicate with the operating system through the SMARTS layer.

SMARTS on BS2000/OSD Basics

SMARTS implements a server runtime environment for the execution of the Web I/O Interface server. Technically, SMARTS represents a C runtime system and implements a nearly full-blown POSIX system. It drives a family of tasks which either process dedicated functionality or process the application payload in parallel-executed worker tasks. The tasks with a dedicated functionality are the main or oc task, the system thread loop task (started as second task), the socket communication task, the pfs task and the sequential file I/O task. The pfs task is optional; it processes all I/O operations on the POSIX file system (PFS). The PFS is used for Web I/O Interface server work/print file access method, thus allowing the testing of programs which execute access to work or print files.

SMARTS offers a configurable set of resources to process the application workload. These resources are mainly threads and (worker-) tasks. The actual workload is scheduled by the SMARTS kernel using these resources. In case of momentary shortages of one or the other resource, SMARTS is able to queue incoming requests and to roll-out inactive threads.

All data, processed by SMARTS or a SMARTS application, is located in one common memory pool, the data pool. All code modules, except for some smaller bootstrap routines, are loaded as shared code into another common memory pool, the code pool.

The worker-tasks are the processes (TSNs) by which the Natural runtime is executed. The amount of storage requested by Natural is located in the SMARTS threads within the data pool during the execution of a transaction. If the number of sessions to be processed exceeds the number of threads defined, an internal facility is invoked for rolling the threads. Threads rolled out are placed in compressed format in a so-called roll buffer pool which resides in the data common memory pool as well.

Front-End Stub NATRNWO

The multi-user, multi-tasking, front-end stub NATRNWO is launched to initialize a Natural Web I/O Interface server.

The following topics are covered below:

- Stub Description
- Natural System Variables Used

Stub Description

The task executing the server initialization (TMain) basically is the main listener which waits for incoming requests from the Web I/O Interface client. It owns a session directory to manage multiple clients (users) and their corresponding remote Natural sessions. TMain has the task to accept all incoming requests and to dispatch them to other subtasks (TWork). The process is as follows:

- First, a server connection issued by the user on the client side (the Login button of the Web I/O Interface client) connects to TMain to establish a connection.
- Next, TMain inserts the client into its session directory, attaches a new TWork subtask and passes the connection to TWork.
- TWork initializes a new Natural session and starts the specified Natural application program.
- While the application performs I/O requests, TWork intercepts the I/O data and passes them to the Web I/O Interface client for processing the I/O. The I/O reply is sent back to the server and the server continues the application.
- If the application terminates (reaches the NEXT mode), TWork terminates the Natural session and drops the connection to the Web I/O Interface client.

That is, each client owns one subtask TWork on the Natural Web I/O Interface server. This subtask runs a Natural session (and within the Natural session, a Natural application) and remains active as long as the application is running.

Natural System Variables Used

Within a Natural Web I/O Interface server session, the following Natural system variables are used:

- *TPSYS contains SERVSTUB,
- *DEVICE contains BROWSER,
- *SERVER-TYPE contains WEBIO.

Front-End

Under z/OS, the Natural front-end required for a Natural Web I/O Interface server is a Natural batch driver assembled with the option LE370=YES.

Under , the front-end is called (together with the Natural runtime system) by the front-end stub for session initialization/termination, request execution and session roll-in/roll-out.

Under z/VSE, VM/CMS and BS2000/OSD, the Natural front-end required for a Natural Web I/O Interface server is the Natural Com-plete driver NCFNUC that is delivered with the corresponding Natural Version for Mainframes.

The Natural front-end required for executing the Natural sessions under control of CICS is the Natural remote front-end NATCSRFE that is delivered with the Natural Web I/O Interface server. For more information, refer to the *Natural Web I/O Interface Server CICS Adapter* documentation.

Server Monitor

To enable the administrator to monitor the status of the Natural Web I/O Interface server, a monitor task is provided which is initialized automatically at server startup. Using the monitor commands, the administrator can control the server activities, cancel particular user sessions, terminate the entire server, etc. See *Operating the Web I/O Interface Server*.