Operating the Natural SQL Gateway Server

Starting the Natural SQL Gateway Server

Under z/OS:

The Natural SQL Gateway server can be started as a "started task":

```
//NSBSRV  PROC
//SRV      EXEC PGM=NATRNSV,REGION=4000K,TIME=1440,
// PARM=('POSIX(ON)/NSBSRV1')
//STEPLIB  DD DISP=SHR,DSN=NSBvrs.LOAD
//CMPRINT  DD SYSOUT=X
//STGCONFG DD DISP=SHR,DSN=NSBvrs.CONFIG(SRV1)
//STGTRACE DD SYSOUT=X
//STGSTDO  DD SYSOUT=X
//STGSTDE  DD SYSOUT=X
```

- where NSB is the product code and *vrs* is the version, release, system maintenance level number of the Natural SQL Gateway server.

Note:
`PARM=('POSIX(ON)/NSBSRV1') - POSIX(ON)` is required for a proper LE370 initialization, and `NSBSRV1` is the name of the server for the communication with the monitor client.

The name of the started task must be defined under RACF and the z/OS UNIX System Services.

Monitoring the Natural SQL Gateway Server

To enable the administrator to monitor the status of the Natural SQL Gateway server, a monitor task is provided which is initialized automatically at server startup. Using the monitor commands described below, the administrator is able to perform functions such as control the server activities, cancel particular user sessions, terminate the entire server, etc.

The following topics are covered below:

Monitor Communication

To communicate with the monitor

- Use the monitor client NATMOPI.

  See *Monitor Client NATMOPI*.

  Or:
  Use the HTML Monitor Client that supports a standard web browser.

  See *HTML Monitor Client*.

  Or:
  Under z/OS, you can alternatively use the operator command MODIFY to execute the monitor commands described below in the section *Monitor Commands*.

  The output of the executed monitor command will be written to the system log.

  Example:

  ```
  F jobname,APPL=ping
  ```

  sends the command ping to the Natural SQL Gateway server running under the job *jobname*.

Monitor Commands

The Natural SQL Gateway server supports the following monitor commands:

Command Name	Action
ping	Verifies whether the server is active. The server responds and sends the string I'm still up
terminate	Terminates the server.
abort	Terminates the server immediately without releasing any resources.
set *configvariable value*	With the set command, you can modify server configuration settings. For example, to modify TRACE_LEVEL: set TRACE_LEVEL 0x00000012
list sessions	Returns a list of active Natural sessions within the server. For each session, the server returns information about the user who owns the session, the session initialization time, the last activity time and an internal session identifier (*session-id*).
cancel session *session-id*	Cancels a specific Natural session within the Natural SQL Gateway server. To obtain the session ID, use the monitor command list sessions.
help	Returns help information about the monitor commands supported.

Runtime Trace Facility

For debugging purposes, the server code has a built-in trace facility which can be switched on, if desired.

The following topics are covered below:

- Trace Medium
- Trace Configuration
- Trace Level
- Trace Filter

Trace Medium

Under z/OS, the Natural SQL Gateway server writes its runtime trace to the logical system file `STGTRACE`.

Trace Configuration

The trace is configured by a trace level which defines the details of the trace. Once a trace is switched on, it can be restricted to particular clients or client requests by specifying a trace filter, see also Natural SQL Gateway server configuration parameter `TRACE_FILTER`.

Every session is provided with a 32-bit trace status word (TSW) which defines the trace level for this session. The value of the TSW is set in the Natural SQL Gateway server configuration parameter `TRACE_LEVEL`. A value of zero (0) means that the trace is switched off.

Trace Level

Each bit of the TSW is responsible for certain trace information. Starting with the rightmost bit:

Trace Bit	Trace Information
31	Trace main events (server initialization/termination, client request/result).
30	Detailed functions (session allocation, rollin/rollout calls, detailed request processing).
29	Dump internal storage areas.
28	Session directory access.
27	Dump send/reply buffer.
26	Dump send/reply buffer short. Only the first 64 bytes are dumped.
25 - 16	Free.
15	Trace error situations only.
14	Apply trace filter definitions.
13 - 08	Free.
07 - 01	Free.
00	Reserved for trace-level extension.

Trace Filter

It is possible to restrict the trace by a logical filter in order to reduce the volume of the server trace output.

- The filter can be set with the configuration parameter `TRACE_FILTER`.
- The filter may consist of multiple *`keyword=filtervalue`* assignments separated by spaces.
- To activate the filter definition, the trace bit 14 in the trace status word (see *Trace Level*) must be set.

The filter keyword is:

`Client`	Filters the trace output by specific clients.

The following rules apply:

- If a keyword is defined multiple times, the values are cumulated.
- The value must be enclosed in braces and can be a list of filter values separated by spaces.
- The values are not case sensitive.
- Asterisk notation is possible.

Example:

```
TRACE_FILTER="Client=(XYZ P*)"
```

Each request of the user ID `XYZ` and each request of the user IDs starting with a `P` are traced.