Natural as a Server under z/VSE

This document applies under z/VSE only. It covers the following topics:

- Functionality

- Natural Nucleus Installation in a Server Environment

- Print and Work File Handling with External Datasets in a Server Environment

Functionality

Besides being a programming language, Natural can also act as a server in a client/server environment. It can provide services, such as the execution of Natural subprograms. Part of the server functionality is the enhanced batch driver. There are a lot of underlying protocols for the client/server communication, such as the execution of stored procedures for DB2 and the execution of remote procedure calls, see the *Natural Remote Procedure Call (RPC)* documentation.

Natural Server Stub

Natural as a server runs in a separate region or within the server subsystem region, for example for DB2 stored procedures. To run Natural as a server, a service-specific server stub is required. This server stub is supplied as part of the server product. It controls all service requests and is the only interface to the Natural server frontend.

There are different server stubs for DB2, for RPC and for others.

Natural Batch Driver

The Natural batch driver (that is, for example, `NATVSE` under z/VSE) has been enhanced to act as the environment-specific interface component which maintains the Natural server sessions and supplies environment-specific services to Natural. It can be linked to the server stub module or loaded by the server stub as a separate module.

The batch driver is able to create and to control multiple sessions by using storage threads including functionality for thread storage compression, decompression and rollout to external storage devices.

When the batch driver is called by the server stub for the first time (during server initialization), the storage threads are created in main storage. The number and size of the storage threads is determined by the server stub. Then a static Natural session is initialized. This includes profile parameter evaluation and the allocation of static storage buffers. The resulting pre-initialized storage thread is saved in main storage separately. For every new Natural session, this initial 'session clone' is copied into the thread.

When decided by the server stub, a session can be rolled out to be resumed at a later point of time. A roll cache is used by the driver to save the compressed thread storage of a session.

Natural Nucleus Installation in a Server Environment

The Natural nucleus and its batch driver are designed to support both, server and non-server environments. For the server-specific definitions and requirements, please refer to the specific documentation (for example to the *Natural Remote Procedure Call (RPC)* documentation or to the *Natural for DB2* documentation).

You can use a local or a global Natural buffer pool. If you define a local buffer pool, it will be shared by all sessions within the server region.

If a logical print or work file number is to be used for processing within any server session, it must be associated with an access method at session start time. This can be done in `NATPARM` with the macros `NTWORK` and `NTPRINT`, as in the following example, if you want to allow the full range of all print and work file numbers possible:

```
NTPRINT (1-31),AM=STD,OPEN=ACC,DEST=*
NTWORK (1-32),AM=STD,OPEN=ACC,DEST=*
```

The subparameter `DEST=*` defines generic DD name generation during the first `DEFINE WORK FILE` or `DEFINE PRINTER` statement, `OUTPUT` clause (see below). Subparameter `OPEN=ACC` avoids pre-opening of the files at program start time. The open is issued upon the first access of the file.

Print and Work File Handling with External Datasets in a Server Environment

When running many concurrent sessions in one region, there may be resource conflicts with external print and work files. The logical names (DD names) for print and work files are defined by the subparameter `DEST` of macro `NTPRINT` or`NTWORK`, or its dynamic equivalents `PRINT` or `WORK` (defaults `CMPRTnn` and `CMWKFnn`). For normal Natural batch processing, these files are defined in JCL by a logical file and a physical dataset name.

However, DD names are reserved by the operating system for exclusive use by one task, respectively session, that is, if `CMWKF01` is opened by one session for processing, no other session could use this file until it is closed again. Other sessions would get an error if they would try to open it.

In a server environment, all print and work file requests are handled by a dedicated I/O subtask. This ensures dataset integrity and avoids resource contention. It enables the shared usage of print and work files accross Natural session boundaries, that is, multiple sessions can access the same file concurrently.

For exclusive usage of print and work files, Natural offers the following feature to support print and work files in a server environment (both require a special implementation within the Natural application programs for the server environment):

- `DEFINE WORK FILE` or `DEFINE PRINTER` statements, `OUTPUT` clause

The `OUTPUT` clause of these statements can be used

- to define the logical file name for a work or print file, or

- to define the physical dataset name.

If a physical dataset name or a spool file class is specified, the access method itself allocates the dataset dynamically during the execution of the `DEFINE ...` statement. To ensure an unique file name is used, `DEST=*` should be predefined in the `NATPARM` file. This avoids any file name conflicts.