Natural Roll Server Functionality

This document covers the following topics:

- Purpose of the Natural Roll Server
- Natural Version 3.1. Roll Server
- Roll Server in a Single OS/390 System
- Roll Server in a Parallel Sysplex

Purpose of the Natural Roll-Server

With the Natural Roll Server, you can execute Natural in a multiple-address-space system like CICS or IMS/TM; these address spaces may be located within multiple OS/390 images (Parallel Sysplex). You can, of course, also use the Roll Server if you are running a single OS/390 system.

The Roll Server takes control of Natural threads when it receives a read or write request and relinquishes control after the terminal input or output. In this way, the Roll Server reduces physical disk I/Os to the absolute minimum.

The Roll Server runs in its own address space. It provides its services as PC routines. In a Parallel Sysplex environment, an instance of the Roll Server must be started for every single OS/390 image.

What is referred to as Roll Server here consists of two parts:

- the local roll buffer (LRB), a formatted part of the data space,
- the roll file.

This functionality replaces the swap pool used in earlier versions of Natural.

Natural Version 3.1. Roll Server

For the Natural Version 3.1. Roll Server, you need roll files of Version 3.1. The Roll Server version is written to the message log after the Roll Server has been started up successfully. The Version 3.1. Roll Server is downward compatible, that is, it can also run in a Natural Version 2.3 environment.

The list of applied Roll Server zaps is displayed separately if you enter the Natural command DUMP ZAPS. In addition, the list of applied zaps is written to the message log during Roll Server startup.

Note concerning Natural/CICS: The CICS System Recovery Table should include the OS/390 system abend code 0D6.

Roll Server in a Single OS/390 System

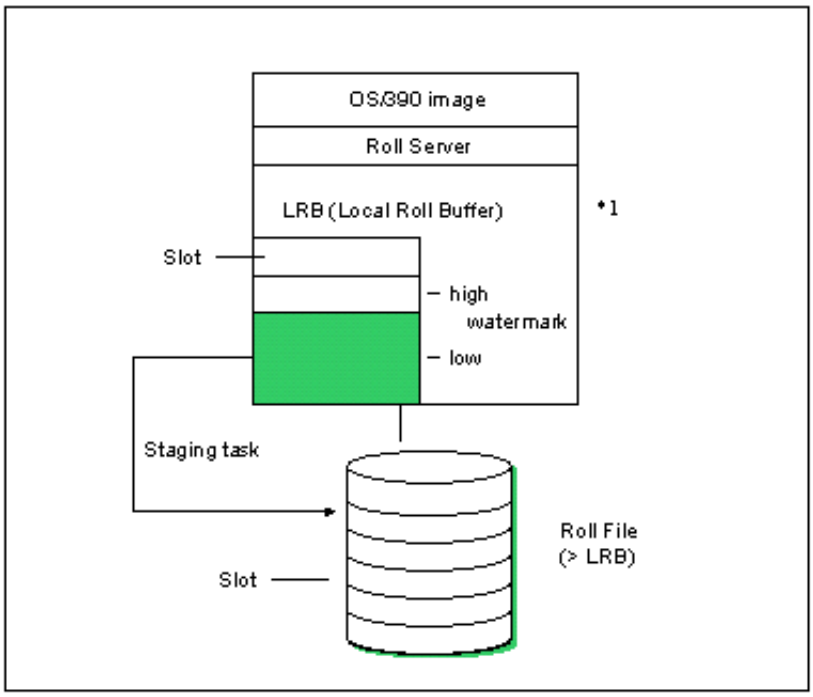
In a single OS/390 system, the Coupling Facility (CF) is not used.

If the Roll Server receives a thread with a write request (terminal output), it checks whether enough space is available in the local roll buffer (LRB). If there is, the thread is written to the LRB. If not, the thread is written to the roll file.

If the Roll Server receives a thread with a read request (terminal input), it checks whether it finds the thread in the LRB. If it does, the thread is read from the LRB and written to the requestor’s address space. If not, the thread is read from the roll file and written to the requestor’s address space.

To ensure that the system performs well and that there is always enough space in the LRB, there are "water marks". If the LRB's high water mark is reached, the staging task is triggered and copies the LRB content to the roll file until the low water mark is reached. Where the high water mark and the low water mark are placed is therefore an important issue of performance tuning. For more information on performance tuning, see the section Roll Server Performance Tuning.

Illustration of how the Roll Server works in a single OS/390 system:



Legend:

* 1 Read requests and write requests reach the Local Roll Buffer. Is there enough space in the LRB? If the LRB is is filled up to the high water mark, start the staging task and write LRB content to the roll file.

Roll Server in a Parallel Sysplex

In a Parallel Sysplex, the Coupling Facility (CF) must be used. There may be up to 32 connections to a Coupling Facility structure, which is the maximum number of OS/390 images in a Parallel Sysplex. Therefore, only one instance per OS/390 image may access one Coupling Facility structure.

If the Roll Server receives a thread with a write request (terminal output), it checks whether enough space is available in the local roll buffer (LRB). Asynchronously, the Roll Server reads the thread from the LRB and writes it to the roll file. If there is enough space, the thread is written to the LRB. If not, the thread has been written to the roll file anyway.

If the Roll Server receives a thread with a read request (terminal input), and the last write request has come from the same OS/390 image, the Roll Server reads the thread direct from the LRB. If the last write request did not come from the same OS/390 image, the thread is read from the roll file and written to the requestor's address space.

The roll file is a BDAM file logically subdivided into directory and fixed-length slots. The slot size is a parameter in the roll-file formatting routine NATRSRFI. Slots must be larger than the largest possible compressed Natural thread.

The local roll buffer is contained in a data space and subdivided into fixed-length slots. LRB slots may be smaller than roll-file slots. When a thread is larger than the LRB slot size, it is written to the roll file direct. The number of LRB slots and their size are Roll Server startup parameters; they are important factors in system performance.

The Roll Server may use up to five different roll files. Each of these roll files is logically connected to one local roll buffer. If there are five roll files, there are five corresponding LRBs. The roll files are accessed by a number of different tasks. So if the roll files are created on different disks and have been served on different channels, the roll files can be accessed synchronously.

Natural users are distributed among roll files according to the following algorithm:

(first four bytes of *roll-server-user-id*) modulo number of roll files + 1

where *roll-server-user-id* is a 16-byte, unique string provided by the Natural interface; for more information, see the corresponding interface section in this documentation.

Example:

There are five roll files and the *roll-server-user-id* is "UF"

```
E4C64040 - 80000000 = 64C64040
64C64040 modulo 5 = 1
```

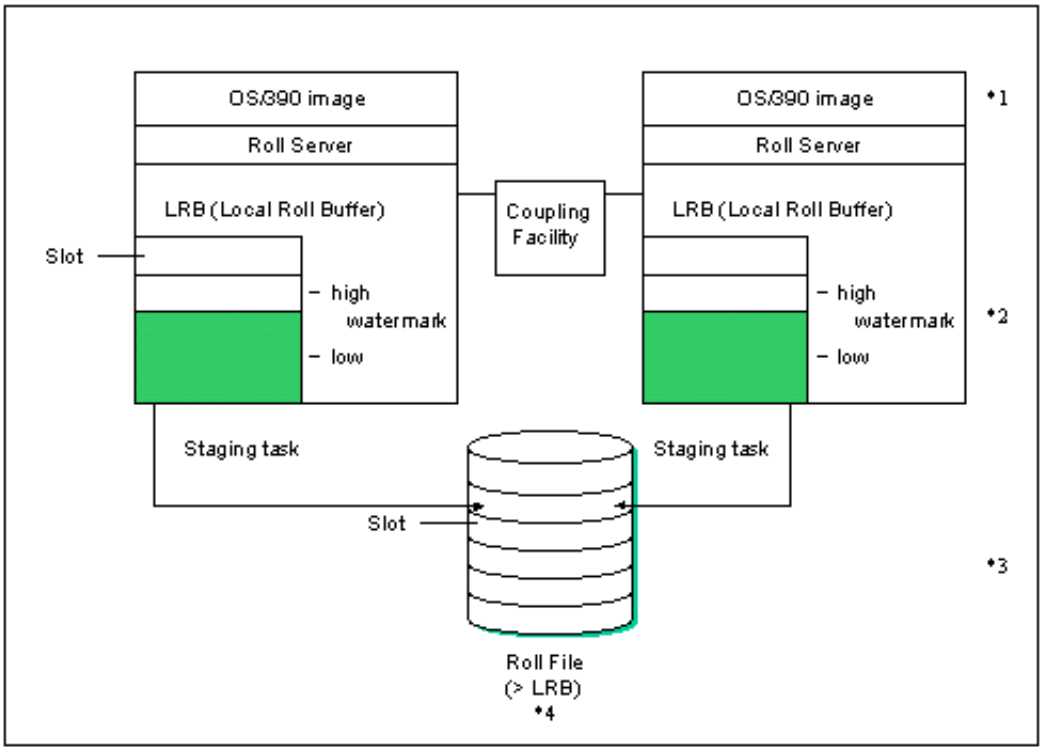
User "UF" is therefore allocated to Roll File 2.

If this algorithm does not guarantee that your user IDs are evenly distributed among the roll files, the Roll Server's user exit NATRSUX1 will help. This is especially relevant in server environments (see Natural as a Server under OS/390), because there, the first eight bytes of the roll server user ID are filled with the name of the server. For more information on this user exit see Natural Roll Server Operation, NATRSUX1 User Exit.

To see how evenly your user IDs are currently distributed, you can use the Roll-Server statistics utility.

To serialize, the Roll Server uses a non-serialized CF list structure without data entries. The Roll Server allocates the structure as a persistent structure. This means that the structure will still be allocated even if all instances of the Roll Server have been terminated. The CF list structure should be defined in the CF policy as non-volatile.

Illustration of how the Roll Server works in a Parallel Sysplex OS/390 system:



Legend:

* 1 - Maximum of 32 OS/390 images
* 2 - Maximum of 32 instances of the Roll Server per one OS/390 image.
* 3 - Maximum of 5 roll files per Roll Server.
* 4 - Read requests and write requests reach the Local Roll Buffer (LRB).
Is there enough space in the LRB?
If the LRB is filled up to the high water mark, start the staging task and write LRB content to the roll file.

