

# Starting a Server

Two methods are supported for starting a server of the SAT product family:

- ONLINE-Start
  - AUTO-START
- 

## ONLINE-Start

The start of a server of any SAT product

- Entire Operations
- Entire Output Management
- Entire Event Management

in any environment supported (OS/390, VSE/ESA, BS2000/OSD) can be performed online. Proceed as follows:

1. Use an online Natural with the following specifications:
  - FNAT must contain SYSSAT library (as installed in Step 2)
  - LFILE 204 must point to your local SYSSAT environment in order to find main member SATP $nnn$  in SYSSATU (see Step 5 and Step 6 of the subsection Installation Procedure).
2. LOGON to the appropriate product library, for which you want to start the server(s).
3. Invoke the product-specific start command (see documentation for the product itself).
4. This start command reads the SATSTART parameter block of the appropriate product in SATP $nnn$  and invokes the server initialization program.
5. You will be informed online about the success of the operation.

## AUTO-START

- For OS/390 and VSE/ESA
- For BS2000/OSD
- For OS/390 and VSE/ESA
- For BS2000/OSD
- Starting Servers with TYPE=SUBTASK
- Starting Servers with TYPE=BATCH

With AUTO-START you can automatically start one or more servers at Entire System Server startup time. Proceed as follows:

1. Link a suitable Natural for this purpose (as described in Step 3 and Step 4):

### For OS/390 and VSE/ESA

This must be a subtask-Natural, because it runs in the address space of the Entire System Server.

### For BS2000/OSD

This must be a multi-user Natural.

2. To activate this process, adapt the startup parameters of Entire System Server. This consists of the following actions:

- Specify the name of the Natural module which should be given control.
- Specify the LOGON commands to invoke program SATSTART in library SYSSAT.

## For OS/390 and VSE/ESA

Parameters to be filled in are marked with brackets <>; required values are denoted by capital letters.

```
NATSHARE=<Name of Natural shared nucleus, if used>
NATNUMSUB=<Maximum number of subtasks> (recommended: 20)
NATMOD=<Name of Natural subtask module as linked in Step 4
STRNTNP1=STACK=(LOGON SYSSAT,<NSC-USERID>,<NSC-PASSWORD>;
STRNTNP2=SATSTART;FIN),AUTO=OFF
```

or, if Natural Security is not installed:

```
STRNTNP1=STACK=(LOGON SYSSAT;SATSTART;FIN),AUTO=OFF
```

## For BS2000/OSD

Parameters to be filled in are marked with brackets <>; required values are denoted by capital letters.

```
JOBNATSUB=<JCL location for Natural subtask AUTO-Start>
PRMNATSUB=<Parameters of ENTER/START-JOB>
NATNUMSUB=<Maximum number of subtasks> (recommended: 20)
```

### Note:

You can find an example of JCL for Natural subtask AUTO-STARTs in the member E.STARTSAT in the LIB.SATnnn library.

3. During startup, the program SATSTART now gets control. As in the case of an online start, SATSTART uses the LFILE setting for File **204** to find its main member.
4. For each SATSTART instruction defined in the SATPnnn program, SATSTART starts a server. The type of the server (batch or subtask) is determined by the parameter TYPE.
  - For TYPE=SUBTASK:  
The Natural subtask module specified with the NATTASK parameter is given control.
  - For TYPE=BATCH:  
The Natural batch module specified with the NATBATCH parameter is given control. The necessary JCL for this batch job is expected in the Natural member specified with the NATSKEL parameter (library is SYSSATU). The jobname of the server task is created automatically.

For more information on the above, see the subsections Starting Servers with Type=Subtask and Starting Servers with Type=Batch.

5. These server sessions can be adapted with the SATENV parameter block:  
Default settings are marked with the prefix SAT. They can be overridden by product-specific parameter blocks. The same holds true for Natural-specific parameter settings (NATENV block).
6. During each server startup, a product-specific initialization module gets control. Its name is automatically derived from parameters given in the SATSTART block in the following way:

<product>SAT<satvers>

For example: NOPSAT23

7. This server initialization module can itself start other servers.
8. You can check the success of this processing either by examining the Entire System Server protocol or by logging on to the online application and testing the server status online.

## Starting Servers with TYPE=SUBTASK

For each SATSTART instruction, in the address space of Entire System Server (OS/390, VSE/ESA), a subtask is started which initiates the server start. The subtask name is built as follows:

pppSTAddddffff

where:

*ppp* = product code  
*dddd* = DBID as specified in the SERVSYSF parameter  
*ffff* = FNR

**Note:**

If you want to start servers as subtasks in a BS2000/OSD environment, proceed as follows:

1. Adapt either the NSBTSKIS member (for ISP format) or the NSBTSKSD member (for SDF format) in the SAT*nnn* source library.

**Note:**

The ADALNK parameter file is optionally supported. To use this function, you must change the member NSBTSKIS or NSBTSKSD correspondingly. Further information is available in the Adabas Release Notes.

2. Assemble it into the Entire System Server load library.

Subtasks are simulated by Entire System Server: batch jobs are submitted under the BS2000/OSD user ID as specified in the ESYUSER parameter. The job names of these batch jobs are built as follows:

pppST*nnn*

where:

*ppp* = product code  
*nnn* = node number

## Starting Servers with TYPE=BATCH

For each SATSTART instruction, a batch job is submitted. For this submit, the user ID specified in the ESYUSER parameter is in effect. The job name is built as follows:

```
pppnnrr
```

where:

```
ppp = prefix as specified in the JOBPREF parameter or product code  
nnn = node number  
rr = run number
```

You must prepare a job skeleton which reflects your system environment and which is used by the SATSTART program. Examples are delivered in SYSSAT which you can use as a basis for your skeletons. Skeletons must reside in the SYSSATU library. You can specify their names with the NATSKEL parameter, for example:

```
NATSKEL=JSKELMVS ! OS/390 environment  
NATSKEL=JSKELVSE ! VSE/ESA environment  
NATSKEL=JSKELBS2 ! BS2000/OSD environment
```