

Common Entire System Server Features

This section describes common Entire System Server Features.

It covers the following topics:

- Running System Automation Tools in Entire System Server
 - Common Diagnostic Features
 - Write-to-Spool for Natural
 - Dynamic Server Management for Entire System Server
 - Zap Reports in Entire System Server
 - Ending Entire System Server
-

Running System Automation Tools in Entire System Server

General

Entire System Server enables the operation of System Automation Tools (e.g., Entire Output Management (EOM), Entire Operations (EOR)) as subtasks in the address space of ESY (OS/390, VSE/ESA) or as pseudo subtasks, i.e., standalone tasks (BS2000/OSD). These System Automation Tools (SAT) are applications on the basis of Natural, which require a Batch-Natural as engine.

SAT products are started by means of ESY startup parameters.

Until now, interaction between ESY and SAT or EOM, EOR, etc. has only been possible on a rudimentary basis, as the operator command SHUTDOWN has been the only command to terminate part or all of the SAT environment.

The configuration of the required Batch-NATURAL task was not very flexible at least under BS2000/OSD.

This section offers an overview of the interfaces between ESY and SAT and deals with the configuration in the overall context.

Activating SAT during Start of Entire System Server

Start of SAT under BS2000/OSD

As Natural subtasks are implemented as separate tasks under BS2000/OSD, the definition of job control instructions is required. The ESY startup parameter JOBNATSUB specifies the location of the SAT-ENTER job. Apart from that, the following can be defined:

- the attributes for the SAT-ENTER job (PRMNATSUB parameter)
- the maximum number of pseudo Natural subtasks (NATNUMSUB parameter)
- and from ESY Version 2.2.2, the input control of dynamic Natural parameters (NATDYNPAR parameter).

The SAT-ENTER job, which is started during the initialization of Entire System Server, reads initialization data and starts the configured SAT products according to the set up definitions.

In general, a distinction must be made between the start of the SAT products via the macros SATSTART TYPE=BATCH and SATSTART TYPE=SUBTASK. To obtain a complete interaction of the SAT products with ESY, the SATP member (see SAT Installation and Customization for details) for the SATSTART macros should always use the TYPE=SUBTASK type. This ensures that both control functions and the Entire System Server shutdown interact with the SAT subproducts. TYPE=BATCH jobs are not known to Entire System Server.

The products started by SAT (e.g., EOM, EOR) run via separate ENTER jobs. In case of SATSTART TYPE=SUBTASK a job-skeleton is used for these ENTER tasks, which in the past had to be part of the ESY module library in object module format. From ESY 3.1.1, this job skeleton is definable as part of the ESY startup file.

Recently Introduced Startup Parameters under BS2000/OSD

- ESY 2.2.2: NATDYNPAR startup parameter for the operation of SAT products

Customers using SAT know the problems with the interaction between the Batch-Natural and the job skeleton used to generate the ENTER file for the products started by SAT. To keep the configuration as variable as possible, the ESY startup parameter NATDYNPAR was introduced to enable the user to read the dynamic Natural parameters via FILE, SYSDTA or SYSIPT according to the generation of the Batch-Natural driver. This assures that the dynamic parameters are correctly processed by the Batch-Natural.

- ESY 3.1.1: NATURAL-SUB-TASK job skeleton as part of the ESY startup parameter file

Starting with ESY 3.1.1, the above complicated method is not used any longer, which consisted of delivering the ENTER file job skeleton of the products started by SAT as a DO procedure with integrated assembler source. This DO procedure was executed during installation and generated an object module, which was loaded in the ESY view processor NATURAL-SUB-TASK and which was modified before execution by means of the corresponding parameters. This resulted in an ENTER job, which started an EOM or EOR task.

Now, the job skeleton is a component of the ESY startup parameter file.

The jobs P.NSBTSKIS and P.NSBTSKSD are still delivered as ESY source library elements, but they are only included for compatibility reasons.

The job skeleton can be defined at any location in the ESY startup parameter file, but it must be started using the keyword SATSKEL-BEGIN and terminated with the keyword SATSKEL-END.

The following abridged example of an ESY startup parameter file shows its use:

```
NODE=113
TIME=30
... more parameters ...
JOBNATSUB=$NPR.E.SAT.113
PRMNATSUB=RESOURCES=*PAR(CPU-LIMIT=*NO)
NATDYNPAR=FILE
NATNUMSUB=20
*
SATSKEL-BEGIN
/.&UID LOGON
... more JCL ...
/ LOGOFF SYS-OUT=DEL
SATSKEL-END
```

A complete example is part of the delivery files. A comprehensive description is provided in the section Startup Parameters.

Control of SAT during Entire System Server Operation

Starting with ESY 3.1.1, the NATPROC-USERS view contains an extension of the LIST function. If the field FULL-SCAN = YES, the view processor also lists all internal tasks in addition to the ESY users. This simplifies the control of the tasks controlled by ESY.

Activating/Deactivating NATURAL-SUBTASKS (SAT) during Operation

The operator command SHUTDOWN allows communication between ESY and the SAT product specified by using the operand and communicates the termination request. Communication is carried out by using the view processor EVENTING. By using this view, the SAT products obtain all required information, which ESY has passed by using the normal user interface.

New with ESY 3.1.1 is the operator command START ALL to restart SAT. It restarts the entire SAT environment without restarting ESY. First the SAT task is restarted and subsequently all defined SAT products are restarted. START ALL can be used only if the entire SAT environment has been stopped on its own or by operator command SHUTDOWN ALL. These two commands enable the user to have a "yo-yo" or "bounce" during normal operation.

SAT configurations can be corrected and tested while ESY is up and running.

Note that the operator command SHUTDOWN can address individual SAT products via parameters, but that the START command only accepts the ALL parameter.

Deactivating SAT during Entire System Server Stop

Special processing is required for the shutdown of ESY, when the SAT products have been started. The NATURAL-SUB-TASKS must be informed of the imminent termination. This is carried out by means of the view processor EVENTING. Having communicated the termination information, ESY checks the status of the NATURAL-SUBTASKS over short intervals. If they have terminated on their own, shutdown handling in ESY will be continued. In the meantime, user requests are still processed, as if the shutdown command had not been issued.

The Deferred Shutdown mechanism limits the time interval between the command for termination of the Entire System Server and the actual ESY termination. It would be possible, however, that the SAT products may not have accepted the shutdown request for various reasons or that they are busy with termination for an unusual period of time.

To eliminate this problem, the startup parameter SHUTDOWN-MAX-DELAY is available starting with ESY 3.1.1. This parameter limits the Deferred Shutdown to a specified number of seconds. If the time limit is reached, Entire System Server will terminate without properly closing down the SAT tasks.

If this situation occurs, why the SAT products did not stop within the defined time interval must be checked. In this case, Software AG support should be consulted, if necessary. As the monitors implemented in EOM or EOR have wait cycles, SHUTDOWN-MAX-DELAY=180 should be used initially. If all NATURAL-SUB-TASKS are stopped, the ESY termination will be continued immediately.

Common Diagnostic Features

There are two common diagnostic features:

- Command Logging
- Creating Trace Data in Entire System Server

Command Logging

For Adabas and Review users, Entire System Server provides extensive command analysis. This is achieved by setting the appropriate startup parameter. The log information is written to the specified dataset via a CLOG DD or FILE statement (see also the Section Startup Parameters). For further information, refer to the appropriate Adabas and Review documentation.

Note:

Entire System Server does not provide a User Exit 2 to manage a full CLOG dataset. Therefore, activate CLOG logging only for testing.

Using Review

To use Review, you must allocate a CLOG file and provide it in JCL with Entire System Server's started task. You must use the startup parameters UEX4=RAOSESYX,LOGGING=YES and set all logging parameters to YES to log all Adabas buffers. We recommend using Review 4.1.3.

Creating Trace Data in the Entire System Server

Introduction

Proper error analysis requires the logging of internal product information. Beginning with Version 3.1.1, ESY is able to generate data that can supply additional diagnosis information to Software AG support and development.

The secondary goal of implementation was that the overall behavior of the Entire System Server not be affected by trace and that only a minimal performance impact would occur. Therefore, trace data are recorded in raw format in order to save time. The data formatting itself is carried out in the diagnosis program ESYTRACE, designated explicitly for this purpose.

Trace data include information such as program start and end, return codes of individual functions and events during request processing. This basic level of information will be supplemented in future versions by new trace data points.

The primary function of the trace is to provide information about the operation logic in case of errors. It is not intended as a log of requests during normal operation; the Adabas Command Log provides more detailed information regarding requests and responses.

Trace Preparation

The trace environment of Entire System Server is activated by using startup parameters. This environment consists of a data buffer, in which the trace information is stored in a wrap-around method.

The data buffer size may be configured by using a startup parameter. This buffer is available externally to the ESYTRACE program running in MONITOR-MODE (please see the description of ESYTRACE for more information). Under VSE/ESA, an additional step is required to use the MONITOR mode of ESYTRACE. Please see the Installation for VSE/ESA documentation for further details.

The following startup parameters are part of the Entire System Server trace facility:

- TRACE to determine
 - whether the TRACE is to be activated (YES);
 - whether the TRACE environment shall only be initialized without starting the TRACE procedure (PREP);
 - or whether TRACE is not to be activated (NO)
- TRACE-LEN to set up the TRACE data buffer size
- TRACE-SAV to determine whether the available TRACE data are to be saved in a file, at normal termination of ESY or in case of ABEND.

The general recommendation for the operation of Entire System Server is to define the parameter TRACE=PREP to reserve a trace buffer (default 8K) and to set TRACE-SAV=YES. Then TRACE can be switched on or off at will by way of an operator command.

For TRACE-SAV, definition of a disk file is required. Under OS/390, a TRACE DD statement is required. Under VSE/ESA, a TRACSAV DLBL (preferred) or TLBL is required, and it is assigned to SYS021. The data are fixed length, 288 bytes. Under BS2000/OSD, assignment is via the logical file name TRACE (e.g., under OSD: /SET-FILE-LINK LINK-NAME=TRACE,FILE-NAME=name); the data are written sequentially in variable record format with a length of 4K as a maximum.

The following list supplies information about the file attributes:

Option	Explanation
BS2000/OSD	Size: SPACE=(4,4) Features: BLKSIZE=(STD,2),FCBTYPE=SAM,RECFORM=V
OS/390	DCB attributes of RECFM=FB, LRECL=288
VSE/ESA	SAM (may be VSAM-managed), fixed length 288 bytes

During startup of Entire System Server, the file is opened and checked for validity. If errors occur during this check or during creation of the file buffer, tracing will be deactivated and a corresponding operator message will be issued on the console.

Trace Activation/Deactivation during Operation

If Entire System Server was started by means of TRACE=PREP or TRACE=YES and the initialization was successful, tracing may be switched on or off by operator command during operation. To do this, the commands TRACE=Y or TRACE=N are available. This enables the tracing to be limited to specific circumstances. A trace stopped with TRACE=N can be switched on again at any time by means of TRACE=Y.

Trace Data Evaluation Using the Diagnosis Program ESYTRACE

As mentioned above, trace data is stored unformatted. A separate program, ESYTRACE, is supplied which formats the raw data. It reads these data in the trace buffer during operation or reads the file created by TRACE-SAV=YES, which is written at termination or ABEND of the Entire System Server and which represents a mirror image of the trace data available at the time of termination.

The prepared result data can both be written to a file and/or to SYSPRINT (OS/390), SYSLST (VSE/ESA) or SYSOUT (BS2000/OSD). Output is written to an optional file - TRACEOUT DD (OS/390), TRACOUT (SYS002) DLBL or TLBL (VSE/ESA) or TROUT (BS2000/OSD).

ESYTRACE is controlled by parms, which are passed to it via the PARM parameter of the JCL/JCS EXEC statement (OS/390, VSE/ESA) or RDATA (BS2000/OSD). The following parameters are supported:

- Position parameter 1: NODE-ID of the Entire System Server or -f or --file

With Online-Trace, the numerical NODE-ID is specified to indicate MONITOR-MODE, whereas -f or --file causes the dataset assigned using the DDNAME TRACEIN (OS/390), TRACIN (SYS001) DLBL or TLBL (VSE/ESA) or logical file name TRIN (BS2000/OSD) to be processed as input (FILE-MODE).

- Keyword parameters:

- -d | --displ

The prepared result data are transferred to SYSPRINT (OS/390), SYSLST (VSE/ESA) or SYSOUT (BS2000/OSD).

- -n | --ntrout

If this parameter is specified, no output file will be generated with the prepared result data using the logical file name TROUT. This key is accepted only if the parameter -d | --displ has also been specified.

- -p | --poll

This option causes ESYTRACE in the MONITOR-MODE not to terminate the program at the end of the buffer, but to wait for further data in the trace buffer. This enables a running ESY node to be monitored in real time. Under VSE/ESA, additional steps are required before this can be used.

For the format of the optional output file, the following file attributes are required:

Option	Explanation
BS2000/OSD	Size: SPACE=(4,4) Features: BLKSIZE=(STD,2),FCBTYPE=SAM,RECFORM=V
OS/390	LRECL=315, RECFM=FBA
VSE/ESA	LRECL=315

The following parameters are recommended for the two modes:

- parameter list for MONITOR-MODE: "NODE-ID --displ --poll"
- parameter list for FILE-MODE: "--file"

The output file generated is useable only for error analysis by Software AG support. It contains no user data whatsoever other than the USER-ID.

For real-time monitoring of error situations, these steps should be used:

1. Start Entire System Server
2. Start diagnosis program ESYTRACE in MONITOR-MODE with the **--poll** option. The **--displ** option is ignored if specified, as the transfer to SYSOUT is redundant to the output in the file assigned via TROUT.

ESYTRACE will format trace data as soon as it is placed in the buffer.

ESYTRACE Diagnosis Program Termination in MONITOR-MODE

When in MONITOR-MODE, ESYTRACE can only be terminated by means of the operator command QUIT. Issue the QUIT command as follows:

Option	Explanation
BS2000/OSD	/INTR tsn,QUIT
OS/390	F stcname,QUIT
VSE/ESA	MSG xx,DATA=QUIT

Write-to-Spool for Natural

(This feature is available under OS/390 and VSE/ESA only.)

- Purpose
- Using the Write-to-Spool Feature
- Additional Notes
- Installation Considerations

Purpose

The Write-to-Spool feature enables Natural users to write reports to the system spool directly. It can be used in any Natural environment (Com-plete, TSO, CICS, Batch, etc.) and uses the Entire System Server view WRITE-SPOOL.

Under OS/390, the SYSOUT is part of the Entire System Server job stream.

Under VSE/ESA, the SYSOUT is spooled to the POWER queue under the job name of the executing Entire System Server nucleus.

From the system pool, users can print their jobs on any local or remote printer, using any software at their site that handles SYSOUT printing.

Important: This feature cannot be used together with **Natural Advanced Facilities**.

Using the Write-to-Spool Feature

To enable this access method, the system spool must be defined as printer in the Natural parameter module (NATPARM) in the following formats:

```
NTPRINT (n),AM=NAF,NAFSIZE=1,...
```

where *n* is a number within the range of defined printers.

Note:

This entry in the NATPARM module must **not** be overwritten dynamically at the start of a Natural session.

Users must define the JES/POWER destination under the OUTPUT class using the DEFINE PRINTER statement in their programs. For example:

```
DEFINE PRINTER (n) OUTPUT 'LOCAL' /* For printing on local JES/POWER printers
```

or:

```
DEFINE PRINTER (n) OUTPUT 'ANYDEST' /* For any valid JES/POWER destination name
```

where *n* is the number in the PRINTER entry in the NATPARM module as described above.

Reports can now be written to the system spool using either of the statements:

```
DISPLAY (n), or
WRITE (n)
```

where *n* is the number in the PRINTER entry in the NATPARM module as described above.

Additional Notes

1. Users can set the output form and number of copies using the FORMS and COPIES clauses of the DEFINE PRINTER statement.
2. The output form can be set using the stroke (/) in the OUTPUT parameter of the DEFINE PRINTER statement. The text before the stroke is taken as the destination code, the text after the stroke is taken to be the form code, for example:

```
DEFINE PRINTER (2) OUTPUT 'DST/FORM'
```

3. Defaults for items such as Entire System Server node, forms and output class are found in the module NATWSPDF. To change these defaults, modify the member accordingly, assemble (and link-edit in OS/390), then relink the Natural nucleus.

Installation Considerations

To enable the Write-to-Spool feature at your site, relink the Natural module as follows:

For OS/390:

```
REPLACE NATPCPS
  INCLUDE NATLIB(nnnnnnn)      An existing Natural nucleus module
  INCLUDE NPRLIB(NATWSP23)     The Write-to-Spool access method for Natural
  INCLUDE NPRLIB(NATWSPDF)    Write-to-Spool defaults
ENTRY      CMSTART
NAME      nnnnnnn(R)
```

For VSE/ESA:

```
PHASE nnnnnnn                Name of Natural nucleus
  INCLUDE .....              All the standard INCLUDE statements for your
  .....                      Natural module, except the NAF modules
  INCLUDE NATWSP23           The Write-to-Spool access method for Natural
  INCLUDE NATWSPDF           Write-to-Spool defaults
ENTRY      CMSTART
```

Important: If you use a shared Natural nucleus, you must link NATWSP23 and NATWSPDF to your shared nucleus.

Dynamic Server Management for Entire System Server

(This feature is available under BS2000/OSD and VSE/ESA only.)

This section covers the following topics:

- Dynamic Server Management Concepts
- Configuring a Dynamic Server Environment
- Activating/Deactivating Dynamic Server Management During Operation
- Controlling Dynamic Server Management During Operation

Dynamic Server Management Concepts

Unlike OS/390, under BS2000/OSD and VSE/ESA, a fixed number of server tasks/subtasks are started in Entire System Server in order to process user requests. Under OS/390, there is a 1:1 relationship between user and subtask, i.e., there is one subtask started for each user.

Use of a fixed number of server tasks/subtasks is due to operating system architecture, as under both BS2000/OSD and VSE/ESA the main Entire System Server task may create only a limited number of tasks/subtasks.

A dispatcher gives user requests to individual servers. The servers use the security profile of the user and process the request accordingly. Under BS2000/OSD, the server with the least load will process the new request. Under VSE/ESA, the first available server will process the request. If none are available, the request will wait until a subtask is free.

However, when using a fixed number of tasks/subtasks, problems may occur. This design cannot adequately react to different workloads. In case of many requests, the servers may not be able to process the requests quickly enough, resulting in delays. Likewise, during idle times Entire System Server cannot release resources.

For this reason, the Dynamic Server Management (DSM) was created. From ESY Version 3.1.1 (BS2000/OSD and VSE/ESA), it enables systems programming staff and operations to dynamically control servers, i.e., to start or stop them as necessary. A minimum number of servers are started during initialization of Entire System Server, which will be automatically increased in case of a large number of user requests up to a definable upper limit. Delays during request processing can thus be avoided to a large degree.

In case of idle times, servers are stopped until a definable lower limit is reached.

The dispatcher mentioned above determines if incoming requests can be processed with the current server configuration, and increases the number of servers if necessary.

A monitor function checks at regular intervals whether too many servers are idle and stops tasks/subtasks as necessary.

Under BS2000/OSD, the most important aid to recognizing bottlenecks is the queue depth as measured by the dispatcher, i.e., how many requests are pending for processing by a server. If this value reaches a definable limit, another server will be started and selected for processing the new request. Under VSE/ESA, new tasks are created when a new request is received and all active servers are busy with other requests (this will change in a future release).

Idle times are measured based on when a server is finished with a user request. If the monitor finds that the idle time exceeds that specified at startup, the server will be terminated.

Configuring a Dynamic Server Environment

For Dynamic Server Management, various new startup parameters are required to explicitly switch on this operating mode and define basic data for its functionality.

Up to now, the startup parameter NUMTASK defined the number of server subtasks for the conventional static operating mode which is still available of course. If the new startup parameters are not used, the dynamic operating mode will remain inactive.

The following startup parameters are available for Dynamic Server Management:

Startup Parameters	Explanation
SERVER-DYN	Defines whether Dynamic Server Management is to be switched on (YES) or off (NO) at initialization. This may be changed via an operator command at any time.
SERVER-MAX	Defines the upper limit of the number of servers. This may be changed via an operator command at any time.
SERVER-MIN	Defines the lower limit of the number of servers. This may be changed via an operator command at any time.
SERVER-NONACT	Fixes the maximum idle time of a server. If the value is reached or exceeded, the server will be terminated if SERVER-MIN is smaller than the current number of servers.
SERVER-QUEUE-DEPTH	Fixes the depth of the task queue; if the server with the least load has reached this value, another server will be started if SERVER-MAX is greater than the current number of servers.
SMFTIME	VSE/ESA only - how often the monitor task will check for idle servers.

In dynamic operating mode, the startup parameter NUMTASK is used to define the initial number of servers. The following relationship exists between the parameters for the definition of the initial, upper or lower limit of running server tasks/subtasks:

SERVER-MIN <= NUMTASK <= SERVER-MAX

The general recommendation for the operation of Entire System Server in dynamic operating mode is to define the parameter SERVER-DYN=YES and to leave all other parameters at their default values. If the default values are not sufficient under BS2000/OSD, you are recommended to decrease the value for SERVER-QUEUE-DEPTH or to set it to 1, if necessary, so that there will not be any queues. However, the value of SERVER-MAX may limit the number of new servers started.

Activating/Deactivating Dynamic Server Management during Operation

For simplified Dynamic Server Management administration, the operator command interface was enhanced so that most of the startup parameters can be set dynamically as well.

The values for SERVER-DYN, SERVER-MAX, and SERVER-MIN can be modified.

It is also not required to explicitly prepare the dynamic server environment at the time of initialization of Entire System Server. It may be activated by operator command at any time.

Controlling Dynamic Server Management during Operation

The operator command `SERVERS` lists the information about the server load to reflect the current state of Dynamic Server Management.

The following output was generated during a test operation under BS2000/OSD:

```

11:21:31 ADAI29 00113   OPER CMD: SERVERS
11:21:31 XCO0016I 00113 Operator typed in: SERVERS.
11:21:32 ESY0308I 00113 SERVER STATUS VP CMD USER ACT IDLE VIEW.
11:21:32 ESY0308I 00113 -----
11:21:32 ESY0308I 00113 2EF8 ACTIVE 2 2 DC1 Y 67 2.
11:21:32 ESY0308I 00113 2EGK ACTIVE 2 2 DC1 Y 48 2.
11:21:32 ESY0308I 00113 2EGL ACTIVE 0 0 ETB N 10 190.

```

It indicates that Entire System Server currently operates three server tasks. The servers with the TSN 2EF8 and 2EGK are active, while Server 2EGL has not been used for 10 seconds (for details, see also the Section Operator Commands). The display under VSE/ESA is similar, with the exception of the values under the SERVER column.

The view `NATPROC-USERS` also returns information about the state of internal tasks - i.e., the server as well - so that server control by means of a program is possible.

ZAP Reports in Entire System Server

Starting with Version 3.1.1, Entire System Server prints a report of all applied zaps at ESY startup. This information is determined during startup and is written to `DDNAME SYSPRINT (OS/390)`, `SYSLST (VSE/ESA)` or `SYSLST01 (BS2000/OSD)`.

Note for BS2000/OSD:

Each ESY task except the console task generates this ZAP report on `SYSLST01`. If `SYSLST01` is not assigned, no report will be created.

The following sample listing illustrates the report layout. It was created during tests under BS2000/OSD.

OVERVIEW OF APPLIED ZAPS

```

XC41001 XC41002 XC41003 XC41004 XC41005 XC41006 XC41007 XC41008 XC41009
XC41010 XC41011 XC41012 XC41013 XC41014 XC41015 XC41016 XC41017

```

CSECT Mapping

CSECT	EP	DATE	TIME	ZAPS
NPRINIT	01000000	2000-09-18	10:37:33	XC41001 XC41002 XC41003 XC41004 XC41005 XC41006 XC41007 XC41008 XC41009 XC41010 XC41011 XC41012 XC41013 XC41014 XC41015 XC41016 XC41017
CHKLINK	01000E58	2000-09-18	10:32:30	NONE
CMDX2	010012A0	2000-09-18	10:32:38	NONE
GETPARMS	01001878	2000-09-18	10:34:36	NONE
LOAD2	01003330	2000-09-18	10:34:57	NONE
NATPCMDL	010038A0	2000-09-18	10:36:15	NONE
NATPNAT	01004180	2000-09-18	10:36:54	NONE
NATPREP	01004A08	2000-09-18	10:37:04	NONE
NATPSRV	010052D0	2000-09-18	10:37:12	NONE
NATPSUBT	01005FD8	2000-09-18	10:37:21	NONE
NATPUSR	01006780	2000-09-18	10:37:28	NONE
NPROPHND	01006A38	2000-09-18	10:37:41	NONE
SCANECET	010076D8	2000-09-18	10:37:56	NONE
SYNCADA	01007F00	2000-09-18	10:38:19	NONE
SYSINFO	01008298	2000-09-18	10:38:26	NONE
TRACE	010086C0	2000-09-18	10:38:34	NONE
WTO	01009438	2000-09-18	10:39:02	NONE
XCOMINIT	01009DF0	2000-09-18	10:39:19	NONE
XCOMMMAIN	0100A3E8	2000-09-18	10:39:28	NONE
XDBOPER	0100B1F8	2000-09-18	10:47:39	NONE
XDBPRSTP	0100B4D0	2000-09-18	10:47:54	NONE
XDBSTOP	0100B758	2000-09-18	10:48:03	NONE
XDBTIME	0100BCC8	2000-09-18	10:48:14	NONE
XCOMNUC	0100BFF0	2000-09-18	10:39:38	NONE
XDBPROC	0100DAB8	2000-09-18	10:47:45	NONE
ANSWER	0100EA90	2000-09-18	10:32:20	NONE

OS/390 and VSE/ESA reports have a similar format.

The first part of the report lists all installed program corrections (OVERVIEW OF APPLIED ZAPS). The second part provides detailed information about the names of the program sections (CSECT), the entry points (EP), the creation date and time (DATE and TIME), and the installed corrections (ZAPS) per program section.

This information may help to get an overview about applied zaps. It is also useful for Software AG support.

Ending Entire System Server

For information on how to terminate Entire System Server, see Section Operator Commands in the Entire System Server Administration Documentation.

For BS2000/OSD, see How to Start/End Entire System Server on BS2000/OSD in Section BS2000/OSD Considerations of the Entire System Server Administration Documentation.

Return Codes Issued by Entire System Server at Termination

On OS/390 and on VSE/ESA

When Entire System Server terminates due to reasons other than an ABEND, a return code is issued. A return code **0** indicates no abnormal incidents occurred during the run. A return code **4** indicates that a subtask ABENDED at some time during the run; check the JES job log for details. A return code **8** indicates that Entire System Server never started due to a bad parameter or other reason; check the JES job log for details.

On BS2000/OSD

At program termination, the Entire System Server components set a return code, which is transferred to a monitoring job variable. The status display for successful execution is C' \$T 0000', the status for abnormal termination is C' \$A 0008'.