software AG
A SOFTWARE GMBH BRAND

System Automation Tools

Version 3.5.3

October 2025

ADABAS & NATURAL

This document applies to System Automation Tools Version 3.5.3 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: SAT-DOC-353-20251001

Table of Contents

Preface

This documentation is organized under the following headings:

Release Notes	What is new in this version of System Automation Tools?
Introduction	What can be done with System Automation Tools?
Installation on Mainframe Platforms	How to install System Automation Tools on z/OS.
Installation on UNIX Platforms	How to install System Automation Tools on UNIX platforms.
Definitions for Natural and ESM Products	How to define the System Automation Tools parameters for Natural and Entire Systems Management products.
Definitions for EntireX Broker Access - SATBKR	How to customize the text object SATBKR to set the parameters for EntireX Broker Access.
Definitions for Entire System Server - SATSRV	How to customize the text object SATSRV to set the parameters for Entire System Server nodes on UNIX and Windows
Starting a Server	How to start a server of an Entire Systems Management product.
Event Store	How to use the event store.
Messages	How to display System Automation Tools messages.

1 About this Documentation

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *folder.subfolder.service*, APIs, Java classes, methods, properties.
Italic	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at **https://documentation.softwareag.com**.

Product Training

You can find helpful product training material on our Learning Portal at **https://learn.softwareag.com**.

Tech Community

You can collaborate with Software GmbH experts on our Tech Community website at **https://tech-community.softwareag.com**. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software GmbH news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at **https://github.com/softwareag** and **https://hub.docker.com/publishers/softwareag** and discover additional Software GmbH resources.

Product Support

Support for Software GmbH products is provided to licensed customers via our Empower Portal at **https://empower.softwareag.com**. Many services on this portal require that you have an account. If you do not yet have one, you can request it at **https://empower.softwareag.com/register**. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

I

2 Release Notes for Version 3.5.3

These *Release Notes* inform you of the enhancements and new features that are provided with Version 3.5.3 of System Automation Tools, and apply to all supported operating systems: z/OS and UNIX (Linux).

Prerequisites for Mainframes

Before you can install System Automation Tools on a mainframe computer, the following Software GmbH products must have been installed at your site:

- Adabas Version 8.5.4 or above;
- Natural Version 9.2 or above; the installation must include the Software AG Editor component;
- Entire System Server, Version 3.7.2 or above;
- Entire System Server for UNIX/Windows, Version 2.1.16 or above (optional, for Linux/Windows access);
- Natural Security (optional).
- EntireX Version 10.9 or higher for z/OS (optional, for multi-node support);
- Entire Net-Work (optional, for multi-node support);

Prerequisites for UNIX

Before you can install System Automation Tools on a UNIX system, the following Software GmbH products must have been installed at your site:

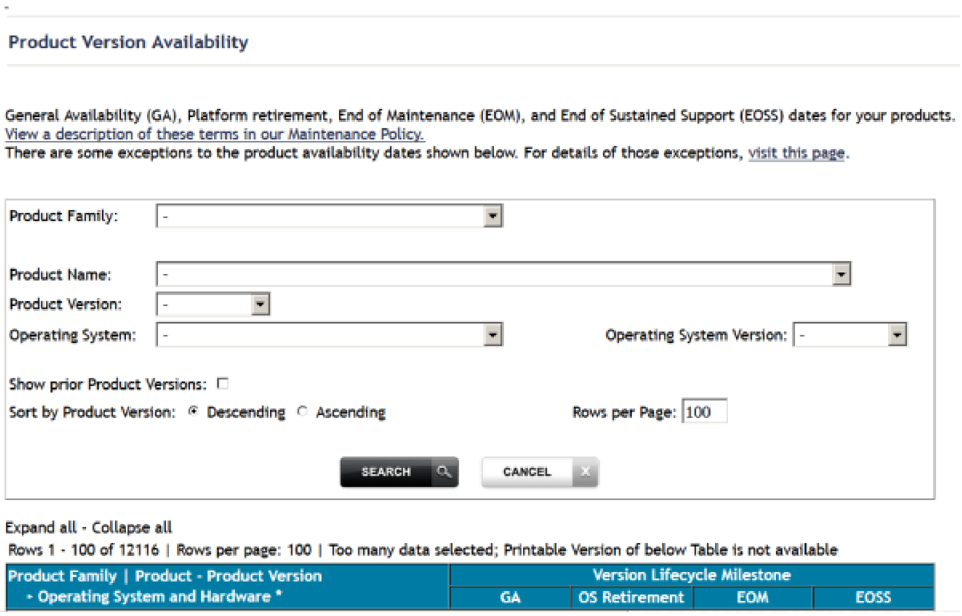
- Adabas for UNIX, Version 7.1 or above;
- Natural for UNIX, Version 9.2 or above;
- Entire System Server for UNIX, Version 2.1.16 or above;
- EntireX Version 10.9 or above (optional, for multi-node support and cross-platform access);

Supported Versions

With the release of System Automation Tools Version 3.5.3, versions below 3.5.2 are no longer supported. Therefore, we strongly advise that you switch to the current version at your earliest convenience.

You can view all available Software GmbH product versions and check the dates when their maintenance ends by visiting the Empower website at ***https://empower.softwareag.com/***:

1. Log in to Empower.
2. Expand **Products & Documentation** in the left menu of the page and select **Product Version Availability**:



Product Version Availability

General Availability (GA), Platform retirement, End of Maintenance (EOM), and End of Sustained Support (EOSS) dates for your products.
View a description of these terms in our Maintenance Policy.
There are some exceptions to the product availability dates shown below. For details of those exceptions, visit this page.

Product Family: -

Product Name: -

Product Version: -

Operating System: - Operating System Version: -

Show prior Product Versions: □

Sort by Product Version: ● Descending ○ Ascending Rows per Page: 100

Expand all - Collapse all

Rows 1 - 100 of 12116 | Rows per page: 100 | Too many data selected; Printable Version of below Table is not available

Product Family | Product - Product Version • Operating System and Hardware *	Version Lifecycle Milestone			
	GA	OS Retirement	EOM	EOSS

3. Select the required filter criteria from the drop-down list boxes and click on the **SEARCH** button.

   A list of supported Software GmbH products that meet the filter criteria is shown. In addition, the end-of-standard-maintenance dates are indicated in the **EOM** column.

   If you mark **Show prior Product Versions**, only product versions that are out of maintenance are listed.

Migrating from Previous Versions

A migration to Version 3.5.3 is possible from Versions 3.5.2, 3.5.1 and 3.4.3 but not from earlier versions.

For more information on migration see *Migrating from Previous Version*

If you want to migrate environments running Entire Operations Version 5.4 as well as Entire Output Management Version 3.4, you have to upgrade Entire Operations *before* you upgrade Entire Output Management because Entire Operations requires System Automation Tools Version 3.5.3 for execution.

Migrating from Version 3.4.4 to 3.5.3

Modified FDT Fields

Short Name	Adabas	Natural	Natural Field
BM	3 P NU	P5	LOG-MICROSECONDS
BN	3 P NU	P5	LOG-MICROSECONDS-INV
BP	5 A NU	A5	LOG-TIME-ZONE

New/Recreated Superdescriptors

Short Name	Adabas	Natural	Natural Field
S2	SUPDE='S2=AB(1,10),AD(1,7),BM(1,3)'	A20	LOG-ENV-TIME
S3	SUPDE='S3=AB(1,10),BB(1,7),BN(1,3)'	A20	LOG-ENV-TIMEINV
S4	SUPDE='S4=AB(1,10),BA(1,2),BC(1,25,BB(1,7),BN(1,3)'	A47	LOG-ENV-TYPE-OBJECT-TIMEINV
S5	SUPDE='S5=AB(1,10),BA(1,2),BB(1,7)BN(1,3)'	A22	LOG-ENV-TYPE-TIMEINV
S6	SUPDE='S6=AB(1,10),AN(1,8),BB(1,7),BN(1,3)'	A28	LOG-ENV-USERID-TIMEINV
S9	SUPDE='S9=AH(1,10),AD(1,7),BM(1,3)'	A20	LOG-NOP-KEY-OT
SA	SUPDE='SA=AH(1,10),AI(1,10),AD(1,7),BM(1,3)'	A30	LOG-NOP-KEY-ONT
SB	SUPDE='SB=AH(1,10),AI(1,10),AJ(1,4),AD(1,7),BM(1,3)'	A34	LOG-NOP-KEY-ONRT

Existing System Automation Tools log files can be adapted by using Adabas Online Services or another Adabas utility.

Solved Problems

All solved problems of the previous version are included in this version.

3 Introduction

This documentation describes System Automation Tools (SAT) used in combination with Entire Output Management, Entire Operations, and Entire Event Management.

System Automation Tools is only available together with these three products. It is used to start the products of the Entire Systems Management (ESM) family whenever Entire System Server is started (AUTO-Start) or on request, using the product's start monitor function.

Parameters can be specified for:

- the System Automation Tools environment itself,
- the Natural environment used,
- each Entire Systems Management product:
  - Entire Operations,
  - Entire Output Management,
  - Entire Event Management.

System Automation Tools can start servers for the above products:

- Independently of the underlying operating system. This means the same parameters are valid in all environments.
- Independently of the product version, even with different product versions in parallel.
- Independently of the version of System Automation Tools. This means compatibility with future versions of System Automation Tools and the products under its control.
- Independently of the mode of operation in which these servers run: They can run as subtasks or separate batch jobs in z/OS.
- Even in multi-node-environments consisting of any number of nodes. Of course, they must be interlinked with Software GmbH's Entire Net-work products.

II Installation on Mainframes

This section describes the installation of System Automation Tools (SAT) on the z/OS operating system:

Notation *`vrs`* or *`vr`*:

If used in this section, the notation *`vrs`* or *`vr`* stands for the relevant version, release and system maintenance level number of a product. For further information on product versions, see Version in the *Glossary* of the *Natural* documentation.

4 Installing System Automation Tools

Installation Jobs

The installation of Software GmbH products is performed by installation jobs. These jobs are either created manually or generated by System Maintenance Aid (SMA).

For each step of the installation procedure described below, the job number of a job performing the respective task is indicated. This job number refers to an installation job generated by SMA. If you are not using SMA, a sample installation job of the same number is provided in the job library on the System Automation Tools installation medium; you have to adapt this sample job to your requirements.

Using System Maintenance Aid

For information on using Software GmbH's System Maintenance Aid (SMA) for the installation process, see the System Maintenance Aid documentation.

5 Contents of Installation Medium

The installation medium contains the data sets listed below for:

The sequence of the data sets is shown in the Software AG *Product Delivery Report* which accompanies the installation medium.

The SAT (System Automation Tools) load/module and source libraries contain modules shared by the Entire Systems Management products.

z/OS

Data Set Name	Contents
SAT*vrs*.JOBS	System Automation Tools installation jobs
SAT*vrs*.INPL	System Automation Tools system libraries (Natural) and error messages
SAT*vrs*.SYSF	System Automation Tools system file
SAT*vrs*.SYSE	System Automation Tools event-store system file

6 Copying the Medium Contents to Disk

z/OS

Copying the Medium Contents to a z/OS Disk

Copy the data sets from the supplied installation medium to your disk before you perform the individual installation procedure for each component to be installed.

The way you copy the data sets depends on the installation method and the medium used:

- If you use System Maintenance Aid (SMA), refer to the copy job instructions provided in the *System Maintenance Aid* documentation.
- If you are not using SMA and want to copy the data sets from CD-ROM, refer to the README.TXT file on the CD-ROM.
- If you are not using SMA and want to copy the data sets from tape, follow the instructions in this section.

This section explains how to copy all data sets from tape to disk.

- Step 1: Copy Data Set COPY.JOB from Tape to Disk
- Step 2: Modify hilev.COPY.JOB on Your Disk
- Step 3: Submit COPY.JOB

Step 1: Copy Data Set COPY.JOB from Tape to Disk

- Modify the following sample job according to your requirements:

```
//SAGTAPE JOB SAG,CLASS=1,MSGCLASS=X
//* ---------------------------------
//COPY EXEC PGM=IEBGENER
//SYSUT1 DD DSN=COPY.JOB,
// DISP=(OLD,PASS),
// UNIT=(CASS,,DEFER),
// VOL=(,RETAIN,SER=tape-volser),
// LABEL=(2,SL)
//SYSUT2 DD DSN=hilev.COPY.JOB,
// DISP=(NEW,CATLG,DELETE),
// UNIT=3390,VOL=SER=disk-volser,
// SPACE=(TRK,(1,1),RLSE),
// DCB=*.SYSUT1
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//
```

where:

`tape-volser` is the VOLSER of the tape, for example: `T12345`,

hilev is a valid high-level qualifier, and
disk-volser is the VOLSER of the disk.

- Execute the job to copy the data set `COPY.JOB` to your disk.

Step 2: Modify hilev.COPY.JOB on Your Disk

- Modify *hilev*`.COPY.JOB` according to your requirements:

  Set `EXPDT` to a valid expiration date, for example, `99365`.

  Set `HILEV` to a valid high-level qualifier, for example, `USERLIB`.

  Set `LOCATION` to a storage location, for example, `STORCLAS=ABC` or `UNIT=3390,VOL=SER=USR123`.

Step 3: Submit COPY.JOB

- Execute *hilev*`.COPY.JOB` to copy single, multiple, or all data sets to your disk.

7 Installation Procedure

Step 1: Load SAT System Files

(Job I050, Step 3700)

This step can be omitted if System Automation Tools is not being installed for the first time.

System Automation Tools (SAT) uses a SAT system file with `LFILE 131`. Use the Adabas load utility to load the file `SAT`*`vrs`*`.SYSF`. The system file is in Version 7 format.

A migration to Version 3.4.3 is possible from Version 3.3.1, but not from earlier versions.

For the ADALOD utility use the following parameters:

Parameter	Value
MAXISN	100000
DSSIZE	10
UISIZE	100B
NISIZE	1000B
ISNREUSE	YES
DSRU	YES

(Job I050, Step 3702)

This step can be omitted if you do not intend to use the **event store**.

The event store uses a SAT system file with `LFILE 84`. Use the Adabas load utility to load the file `SAT`*`vrs`*`.SYSE`. The system file is in Version 7 format.

A migration from a previous version is not applicable as this system file was introduced with the current version.

For the ADALOD utility use the same parameters as listed above.

Step 2: Scratch SAT Library

(Job I051, Step 3700)

If System Automation Tools has been installed before, scratch the `SYSSAT` library using the Natural SYSMAIN utility, and scratch the error messages with the SYSERR utility.

Step 3: Migrating from Previous Version

If you want to migrate environments running Entire Operations Version 5.4 as well as Entire Output Management Version 3.4, you have to upgrade Entire Operations *before* you upgrade Entire Output Management, because Entire Operations requires System Automation Tools Version 3.5.3 for execution.

Migration from Version 3.4.4 to Version 3.5.3 on Mainframes

Job	Steps	Action
I051	3711, 3713, 3715	Migration of system-file structure from Version 3.4.3 to Version 3.5.3.

Step 4: Create Natural Parameter Module

(Job I060: Steps 3700-3710 (Subtask) / Steps 0010-0015 (Batch)

Modify, assemble and link the parameter module for the Natural subtask.

The module must contain at least the following parameters with the documented minimum values. Entries for other products are possible, but not documented here. See the *Natural Parameter Reference* documentation for further information.

Note: To complete these settings, additional parameters may be required for other products. See the product-specific installation procedure.

Macro	Parameter	Explanation
`NTPRM`	`ESIZE=96`	Size of user-buffer extension area.
	`ASIZE=64`	Entire System Server work area size. Choose a value between `48` and `64`. See the minimum values for Entire System Server.
	`CDYNAM=10`	Number of dynamically loaded modules.
	`MAXCL=0`	Maximum number of program calls (no limit).
	`MADIO=0`	Maximum number of DBMS calls (no limit).
	`ETEOP=OFF`	No end-of-transaction at end of program. `OFF` is the default setting.
	`WH=ON`	Wait on Hold, NAT. This is an obligatory setting.
	`IM=D`	Set input mode.
	`MT=0`	Maximum CPU time: no limit.
	`LS=250`	Physical line size. The default setting is `0`.
	`PS=66`	Page size for Natural reports. The default setting is `0`.

Macro	Parameter	Explanation
	RCFIND=OFF	Response Code 113 will be ignored, and processing of the FIND loop will continue by reading the next record. This parameter is mandatory for Entire Operations. It can also be set dynamically.
	RCGET=OFF	Response Code 113 will be ignored, the system variable *ISN will be set to 0, and processing will continue. This parameter is mandatory for Entire Operations. It can also be set dynamically.
	TD=AUTO	Time Differential: Natural compares the physical (store clock) and logical (system environment) machine times and uses the difference between the two as the setting for the TD parameter. For a time change to take effect for Natural (for example, to change time to summer time or back to winter time), it is therefore sufficient to reset the logical machine time. In central Europe, TD=EURO-CET is recommended. See the *Natural Parameter Reference* documentation for other possible TD parameter settings.
NTLFILE	204,*dbid*,*fnr*	LFILE (logical system file) definition: Use the ID of a physical database (*dbid*) and the number of a system file (*fnr*).
NTLFILE	131,*dbid*,*fnr*	System Automation Tools system file access: Use the ID of a physical database (*dbid*) and the number of a system file (*fnr*).
NTLFILE	84,*dbid*,*fnr*	Optional, only required if the event store is to be used. Use the ID of a physical database (*dbid*) and the number of a system file (*fnr*).
NTDB	PROCESS,148	Entire System Server views cataloged to this database ID.
NTBPI	TYPE=NAT,SEQ=*n*,NAME=*bpname*	Global buffer pool definition: *n* is the sequence number of the buffer pool and *bpname* the name of the buffer pool.
	TYPE=NAT,SEQ=*n*,SIZE=*nnn*	Local buffer pool definition: *n* is the sequence number of the buffer pool and *nnn* the size of the buffer pool. If you use a local buffer pool (for Entire Output Management and Entire Event Management only), SIZE must be 512 or higher. The default setting is 256.
NTPRINT	(1-4),AM=STD, OPEN=ACC	Printer definition.
NTWORK	(1-4),AM=STD	Work-file definition.

Step 5: Link Natural Subtask/Batch

(Job I060: Step 3720 - subtask / Step 0020 - batch)

- z/OS

z/OS

Servers for Entire Systems Management products can be started either as subtasks or as separate batch jobs. Therefore, you need either a subtask Natural or a batch Natural.

The following libraries must be used for the linkage:

Library	Description
`//NATLIB DD DISP=SHR,DSN=SAGLIB.NAT`*`vrs`*`.LOAD`	Natural load library.
`//NPRLIB DD DISP=SHR,DSN=SAGLIB.NPR`*`vrs`*`.LOAD`	Supplied Entire System Server load library.

The result of the subtask linkage must be stored in any steplib of the Entire System Server node used and it must be reentrant. This library, like any steplib of the Entire System Server Started Task, must be APF-authorized.

- Take the link job of an existing batch Natural, link the Entire System Server interface to Natural as described in the current *Natural Release Notes* and include the statements listed below.
- Adjust `NATLIB` to your Natural load library, and `NPRLIB` and `SMALIB` to your Entire System Server load library.
- To make Con-nect features available, ensure that the appropriate CNT/TRS modules are included.

For Entire System Server and Natural

The following example applies to Entire System Server together with Natural. It demonstrates how to link the Natural subtask front-end.

`INCLUDE SMALIB(ESYNODTB)`	Entire System Server node table

For Entire Output Management, Entire Operations and Entire Event Management

The following is an example of how to link the Natural subtask front-end:

`INCLUDE SMALIB (NATOS)`	Natural subtask / batch interface.
`INCLUDE` *`natparm`*	Natural parameter module.
`INCLUDE SMALIB (ADALNKR)`	Adabas reentrant link routine.
`INCLUDE NATLIB (SATDTA)`	SAT Calendar (required for all Entire Systems Management products).
`INCLUDE NATEDT`	Software AG Editor.
`INCLUDE NATEDIT`	Program editor and map editor.
`ENTRY CMSTART`	External entry.
`NAME SAT`*`vrs`*`ST (R)`	Name of Natural subtask for System Automation Tools.

Step 6: Load the INPL

(Job I061, Steps 3700)

Load the programs and error messages for System Automation Tools.

Library	File	Contents
`SYSSAT`	FNAT	System Automation Tools programs and error messages.

Natural Security Environment

Define `SYSSAT` as steplib for all Entire Systems Management products.

Non-Natural Security Environment

The `SYSSAT` library is automatically defined as steplib for all Entire Systems Management products.

Step 7: Create Online Natural Parameter Module

(Job I080)

Modify, assemble and link the parameter module for the online Natural (see the jobs `NAT`*`vrs`*`.JOBS` in your Natural environment).

The relevant parameters are listed under Step 4 above.

In addition, the parameter SSIZE=60 (work area size of Software AG Editor) has to be specified in the NTPRM macro.

The macros NTPRINT and NTWORK can be omitted from the online Natural parameter module.

Step 8: Relink All Online Natural Nuclei

All Natural modules, online and batch, which will be used to execute Entire Systems Management functionality (for example, online usage of SYSEOR, SYSNOM, SYSNCL and batch jobs used for Entire Output Management printing, archiving, etc.) must be relinked:

- Link the Entire System Server interface to Natural, as described in the *Natural Installation* documentation.
- Include SATDTA (SAT Calendar function) from the Natural library and product-specific load modules as described in the documentation of the relevant products.
- To make Con-nect features available, ensure that the appropriate CNT/TRS modules are included.

8 Security Definitions

Natural Security

If Natural Security is installed at your site, you have to create the following security definitions in Natural Security:

- Libraries
- User

Libraries

Library	Description
SYSSAT	System Automation Tools application.
SYSSATU	System Automation Tools user library.

User

Define the Natural Security user representing the various servers of the Entire Systems Management products with the user type "person" and the user ID and password identical to the NSCUSER and NSCPSWD parameters described under *Parameter Blocks and Parameters*.

In the user security profile, specify **Private Library** = Yes.

If you define the above libraries as "people-protected", you have to link this user to them.

External Security System

z/OS only

If Entire System Server is installed with an external security system (RACF, CA-ACF2, or CA Top Secret), a user ID identical to the ESYUSER parameter (described under *Parameter Blocks and Parameters*) must be defined in the external security system.

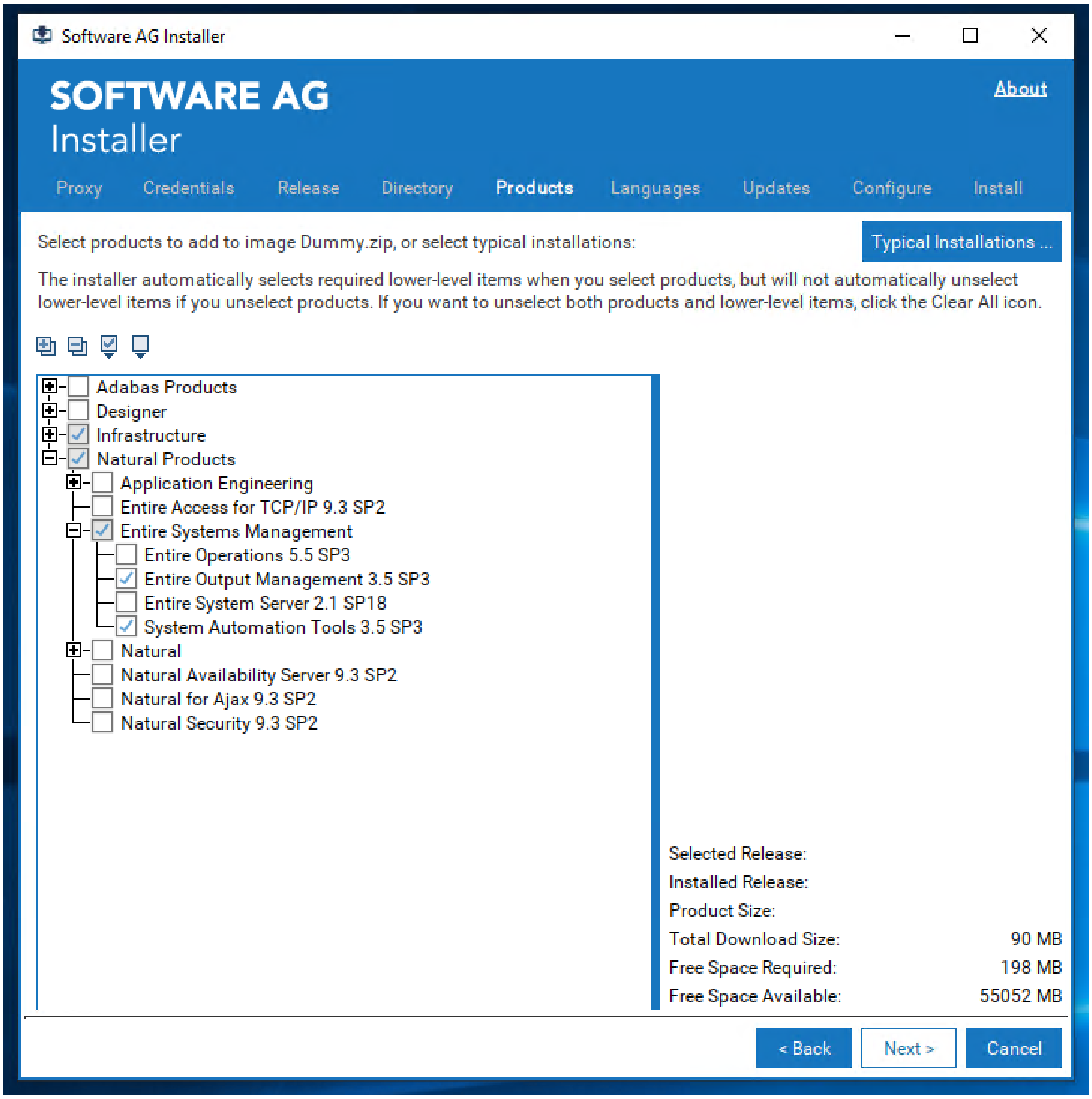
The user must have sufficient authorization to access the spooling system, the console and all data sets used in the online system.

9 Installation on UNIX

This section describes the installation of System Automation Tools on UNIX platforms:

Overview

SYSSAT is a common base-library for the Entire Systems Management Products Entire Operations (NOP) and Entire Output Management (NOM). It will be automatically selected in the Software AG Installer tree, when NOM or NOP is selected for installation..



Select products to add to image Dummy.zip, or select typical installations:

The installer automatically selects required lower-level items when you select products, but will not automatically unselect lower-level items if you unselect products. If you want to unselect both products and lower-level items, click the Clear All icon.

```
⊞ □ Adabas Products
⊞ □ Designer
⊞ ☑ Infrastructure
⊟ ☑ Natural Products
    ⊞ □ Application Engineering
      □ Entire Access for TCP/IP 9.3 SP2
    ⊟ ☑ Entire Systems Management
          □ Entire Operations 5.5 SP3
          ☑ Entire Output Management 3.5 SP3
          □ Entire System Server 2.1 SP18
          ☑ System Automation Tools 3.5 SP3
    ⊞ □ Natural
      □ Natural Availability Server 9.3 SP2
      □ Natural for Ajax 9.3 SP2
      □ Natural Security 9.3 SP2
```

Selected Release:	
Installed Release:	
Product Size:	
Total Download Size:	90 MB
Free Space Required:	198 MB
Free Space Available:	55052 MB

Important Information

Administrator Status

During the installation, the **Sudo** panel may appear. This happens because System Automation Tools depends on the installation of Natural where the **Sudo** panel may be used. For the installation of System Automation Tools, however, you do not need sudoers privileges.

User ID for Installation

System Automation Tools depends on Natural. Therefore, the user ID under which you run the Software AG Installer must not be longer than eight characters. If you use a longer user ID, an error message is shown. You can then exit the installer and use a different user ID or - in case you also want to install other products - return to the product selection tree and deselect Natural.

Installation Directory

During the installation, you are asked to specify an installation directory. The installation of System Automation Tools requires the installation of Natural. If Natural is already installed, choose the directory of your Natural installation. Otherwise, see *Installation* in the Natural for UNIX documentation for detailed information regarding the installation directory. The user that you are using to install must have full read and write permissions to this directory.

File Permissions

The user who starts the installation owns all files that are installed.

The user file-creation mode mask (`umask` command) determines the file permissions for newly created files. Make sure that the `umask` command you are using for the installation will not prevent users from accessing and executing the installed files. On UNIX systems, for example, the command `umask 022` allows full access rights for the file owner and read-only access rights for group members and others.

Side-by-Side Installations

System Automation Tools is a Natural application. Therefore, the rules for Natural also apply for System Automation Tools. For detailed information regarding side-by-side installations, see *Installation* in the Natural for UNIX documentation.

FNAT Usage

By default, a new `FNAT` system file is created in the installation directory during the installation of Natural (`<install-dir>`/*Natural/fnat*). This `FNAT` must always exist, and the global configuration file must have an entry which defines this `FNAT`.

System Automation Tools can only be installed into this `FNAT`.

Note: If you want to check or edit the settings in the global configuration file, use the Natural Configuration Utility.

The Software AG Installer maintains an internal list of installed products, which must coincide with the add-ons that are currently installed in the `FNAT`. This is important for updates and uninstallations to work correctly.

For this reason:

- Do not install products into the `FNAT` without the use of the Software AG Installer.
- Do not replace the default `FNAT` (`<install-dir>`/*Natural/fnat*) with another `FNAT`.
- Make sure to complete the installation of System Automation Tools by using the SYSPCI utility (this is explained later in this documentation).

If an error occurs due to the above-mentioned scenarios, the only way to solve the problem is a new installation. In some situations, one of the following workarounds may help:

- Workaround 1: Complete the previous installation by using the SYSPCI utility.
- Workaround 2: Uninstall the product and then start the installation once more.

Upgrading Your System Automation Tools Environment

When one of the first two digits of the version number changes, we consider an installation as an *upgrade* installation.

Updating Your System Automation Tools Environment

When the first two digits of the version number remain the same and the third or fourth digit changes, we consider an installation as an *update* installation.

Scripts located in the *<install-dir>/System Automation Tools/INSTALL* will not be replaced. Thus, user changes in scripts will be kept. If a script changes with a System Automation Tools update, you can find the updated scripts in the *<install-dir>/System Automation Tools/INSTALL/tpl*directory. The name of an updated script consists of the original name followed by *.tpl*. For example, *nopenv* is then named *nopenv.tpl*. Administrators and users can adapt any scripts manually according to their own needs.

If you want to use the event store, you also have to define LFILE 84 with the SYSPCI utility (see ***Setting Up Your Products Using the SYSPCI Utility***).

Customizing the Application SYSSAT

This menu item contains the creation of the application SYSSAT in your Natural FNAT directory. In addition, the shared library will be copied automatically to NATEXTLIB.

Before you perform this step:

- Make sure that enough disk space is available in the target environment.
- Make sure that you have write access rights to the Natural FNAT directory, as well as to the directory specified by the NATEXTLIB parameter in the local configuration file as described in the *Natural Configuration Utility* documentation.

Continue with the instructions in ***Definitions for Entire System Server -SATSRV***.

Setting Up Your Products Using the SYSPCI Utility

After you have installed your product for the first time, you need to set up a number of files, parameters and individual settings depending on your environment. These are described below. To set them up, you use the SYSPCI utility. For detailed information on this utility, see *SYSPCI Utility - Product Configuration and Initialization* in the *Natural Tools and Utilities* documentation.

Before you can define System Automation Tools system files, the SAT system file (LFILE 131) must be defined.

System Automation Tools requires the following Adabas system files:

File	Logical File Number (LFILE)
`NOP-SYSF1`	216
`NOP-SYSF2`	85
`SAT-EVENTSTORE` (optional, only if you want to use the event store).	84

The database IDs and file numbers of the new or existing files you specify with the SYSPCI utility are entered into the default parameter files System Automation Tools (`NOPPARM`), Entire Output Management (`NOMPARM`, if available) and Natural (`NATPARM`).

The required Adabas files can either be local or remote:

- **Remote Access**

  If the file is located in a remote database, Entire Net-Work must be active and the database must be accessible.

  > **Note:** For Natural Security, see also *Using Natural Security on Multiple Platforms* in the *Natural Security* documentation.

- **Existing Local File**

  Before you start the SYSPCI utility, make sure that the Adabas database containing the required files is active. With this version, you can continue to use your existing files. No migration of data from the previous version to the current version is necessary.

- **New File**

  Before you start the SYSPCI utility, make sure that the Adabas database which will contain the required files is active. The SYSPCI utility will load and initialize these files. This should be also done if another file is required for your product.

Before you create new files with the SYSPCI utility, make sure that the ASSO and DATA sizes of your Adabas database are appropriate for these files. It is therefore recommended that you check the Adabas *.fdu* files in the *<install-dir>/<product>/INSTALL/<product-code>* directory for the used sizes. If required, change your database setup so that the files can be created.

For Natural Security, for example, the ASSO and DATA sizes are not appropriate if you are using the default database. The *.fdu* files for Natural Security can be found in the *<install-dir>/Natural/INSTALL/nsc* directory.

In addition, make sure that the Adabas nucleus parameters listed in the following table are set for the database you want to use at database startup. They are not appropriate if you are using the default nucleus parameters.

`LWP`	Must be at least `1,000,000`.
`OPTIONS`	The option `TRUNCATION` must be set in the `OPTIONS` parameter.

Notes:

1. After Natural Security has been initialized (activated) with the SYSPCI utility, you need to use a Natural Security nucleus to start Natural. The Natural Security nucleus delivered with the Natural Security installation is called “natsec” and is located in the *<install-dir>/Natural/bin* directory. Start Natural Security with *natsec parm=NSCPARM*. Alternatively, you can back up the nucleus called “natural” and rename “natsec” to “natural”.
2. When you have installed Natural Security, you need to start Natural Development Server with a Natural Security nucleus (for example, *natdvsrv –s=natsec*).

Uninstalling System Automation Tools

You uninstall System Automation Tools using the Software AG Uninstaller. For detailed information on how to use the uninstaller, see the *Using the Software AG Installer* guide.

In short: to uninstall System Automation Tools, proceed as follows:

1. Open a command window and go to the *bin* directory of your main installation directory.
2. Run the command `uninstall`. This starts the Software AG Uninstaller.

The following files are not removed:

- All files created by the user, for example, System Automation Tools modules in `FUSER` or parameter files.
- `NOPPARM`.

III

10 Definitions for Natural and ESM Products

General

You can define the run-time environment of your products in one or more Natural text objects in the user library `SYSSATU` for System Automation Tools.

You can specify any object name except the main object name, which must conform to the following naming convention: SP*nnnnn*, where *nnnn* denotes the Entire System Server node (with leading zeros) under which the Entire Systems Management products are AUTO-Started. For a 3-digit Entire System Server node, the System Automation Tools text object can still be named `SATP`*nnn*, where *nnn* denotes the Entire System Server node. We recommend that you migrate to the `SP`*nnnnn* names.

In the main text object, you must specify all parameter values needed to start the products. An asterisk (*) in the first column denotes a comment line. Lines prefixed with `SAT` are treated as default values for System Automation Tools or Natural. They can be overwritten by product-specific values. This means that all occurrences of a parameter are merged when the product is started.

For each occurrence of a `SATSTART` entry a product server is started.

Note: If Software GmbH's integrated application development tool Natural ISPF is installed at your site, you can use the SAT menu to perform this maintenance work and for logging on to any of Software GmbH's solutions in the Entire Systems Management product line. The System Automation Tools menu is provided in the `SAT`*vrs*`.INPL` data set. To make this menu available within Natural ISPF's menu system, simply activate the System Automation Tools subsystem of Natural ISPF. For further information, see the section *System Configuration* in the *Natural ISPF Administration Guide*.

General Layout of a Parameter Block

```
prefix block-identifier [keyword=value,...]
```

where:

Parameter	Description
prefix	`SAT` or compressed product code + prefix as specified in the `SATSTART` instruction.
block-identifier	`SATENV`, `NATENV`, `SATSTART` or product block identifier.
[*keyword*=*value*,...]	Block-specific parameter.

Examples

```
SAT       SATENV    NATTASK=NOPSUBT, NSC=NO
NOP521    NATENV    DU=OFF, FUSER= (9,81)
```

Parameter Blocks and Parameters

Table of Parameter Blocks and Parameters

Parameter Block	Parameter	Description
NATENV	-	All profile parameters supported by Natural are possible.
Product Block	-	See product.
SATENV	NSC=YES/NO	Indicates whether Natural Security is installed or not.
	NSCUSER=	If Natural Security is installed, this is the user ID for logging on to it.
	NSCPSWD=	Password for logging on to Natural Security.
	ESYUSER=	User ID for logging on to Entire System Server, if it is installed, and an interface to an external security system is activated.
	NATTASK=	Name of the Natural subtask module for starting a server as a subtask.
	NATBATCH=	Name of the Natural batch module for starting a server as a batch job.
	NATSKEL=	Job skeleton for starting a server as a batch job.
	JOBPREF=*xxx*	Job name prefix *xxx* to be used for building job names when starting servers as a batch job. For example: EOR in the monitor job name EORMON.
SATENV parameters for RPC servers	NUMTASK=	Number of RPC server replications to be started.
	SRVNAME=	Name of an RPC server which will connect to a broker.
	SRVNODE=	Name of the broker service an RPC server will connect to.
	PROFILE=	Natural parameter profile which is used to start one watchdog task per RPC server.
	CHECK-INTERVAL=	This keyword causes the ping function to RPC servers and the check of Entire System Server eventing (shutdown request) interval to be set to *nnnnn* seconds.
SATSTART	PRODUCT=*xxx*	3-letter product code: NOP, NOM or NCL.
	PREFIX=	PRODUCT and PREFIX are compressed into a prefix which identifies the server-specific parameters.

Parameter Block	Parameter	Description
	`TYPE=BATCH/SUBTASK`	Start server as a batch job or subtask.
	`APPLLIB=`	Name of the Natural library where the product is installed.
	`SERVSYSF=`	Product-specific data file. For each `SATSTART` instruction of one SP*nnnnn* text object, a different data file must be referred to.
	`MEMBER=`	You can specify a member where product-specific parameters are located.

Example

The text object `SP00148` in `SYSSAT` provides an example of a main text object. To use this as the basis for your own object, copy it to `SYSSATU` and adapt it.

In the example below, it is assumed that you are running three Entire Systems Management products (Entire Event Management, Entire Output Management and Entire Operations) as subtasks on Node 148. The parameters of Entire Operations are located in a second text object `NOPPARMS`.

- SAT Environment Settings
- Natural Environment Settings
- Product Environment Settings
- Product Automatic Start
- Example Contents of SP00148 in SYSSATU

SAT Environment Settings

<table>
<tr><td><pre>
SAT SATENV NATTASK=SATvrsST
             NATBATCH=NATvrsBA
             NATSKEL=JSKELVSE
             ESYUSER=NOMMON
             NSC=YES
             NSCUSER=NOMMON
             NSCPSWD=NOMMON
             STEPLIB1=(SYSLIBS,dbid,fnr)
             STEPLIB2=(SYSEXT,dbid,fnr)
             STEPLIB3=(SYSTEM,dbid,fnr)
</pre></td><td>Sets the System Automation Tools defaults for all Entire Systems Management products, here: Entire Operations and Entire Output Management.</td></tr>
<tr><td><pre>
NCLvrs SATENV NATTASK=NSATT08
               NSC=NO
               ESYUSER=NCLMON
</pre></td><td>Overwrites some System Automation Tools values for <code>NCLvrs</code>, <code>NOMvrs</code> and <code>NOPvrs</code> respectively.</td></tr>
</table>

NOM*vrs* SATENV NATTASK=NSATT08
NSC=NO
ESYUSER=NOMMON	

NOP*vrs* SATENV NSC=NO
ESYUSER=NOPMON
JOBPREF=*xxx*	

Natural Environment Settings

If the following parameters are passed to Natural as dynamic parameters, the maximum string length of all parameters must not exceed 250 bytes.

SAT NATENV DU=OFF
MAXCL=0
MADIO=0
MT=0
ID=' '
DC='.'
ETID=''
	Sets the Natural defaults for all Entire Systems Management products.
	It is recommended to use ETID=''. This parameter applies to Entire Output Management only. See the Natural documentation for the valid ETID syntax .

NCL*vrs* NATENV FNAT=(1,5)	Overwrites some Natural values for NCL*vrs*, NOM*vrs* and NOP*vrs* respectively.
NOM*vrs* NATENV FNAT=(9,45)	
NOP*vrs* NATENV FNAT=(9,45)	

Product Environment Settings

NOM*vrs* NOMENV ↩ BS2USER=PROD01	Product environment settings are documented in the corresponding product installation documentation.

Product Automatic Start

SAT SATSTART PRODUCT=NCL
PREFIX=*vrs*
TYPE=SUBTASK
APPLLIB=SYSNCLSV
SERVSYSF=(1,7)
	Specifies that the servers for NCL*vrs*, NOM*vrs* and NOP*vrs* respectively are to be started as subtasks.

`SAT SATSTART PRODUCT=NOM`
`              PREFIX=vrs`
`              TYPE=SUBTASK`
`              APPLLIB=SYSNOM`
`              SERVSYSF=(9,46)`	

`SAT SATSTART PRODUCT=NOP`
`              APPLID=SYSEOR`
`              PREFIX=vrs`
`              TYPE=SUBTASK`
`              APPLLIB=SYSEOR`
`              SERVSYSF=(9,65)`	

Example Contents of SP00148 in SYSSATU

`NOPvrs SATENV NSC=YES,`
`               NSCUSER=NOPMON,`
`               NSCPSWD=HUGO`
	Overwrites some System Automation defaults for `NOPvrs` only.

`NOPvrs NATENV DU=ON`	Overwrites some Natural defaults for `NOPvrs` only.

SATDIR Text Object

You have to define your System Automation Tools environment(s) in the text object `SATDIR` in the library `SYSSATU`.

In a new installation, copy the delivered template `SATDIREX` to `SATDIR` and then modify `SATDIR`.

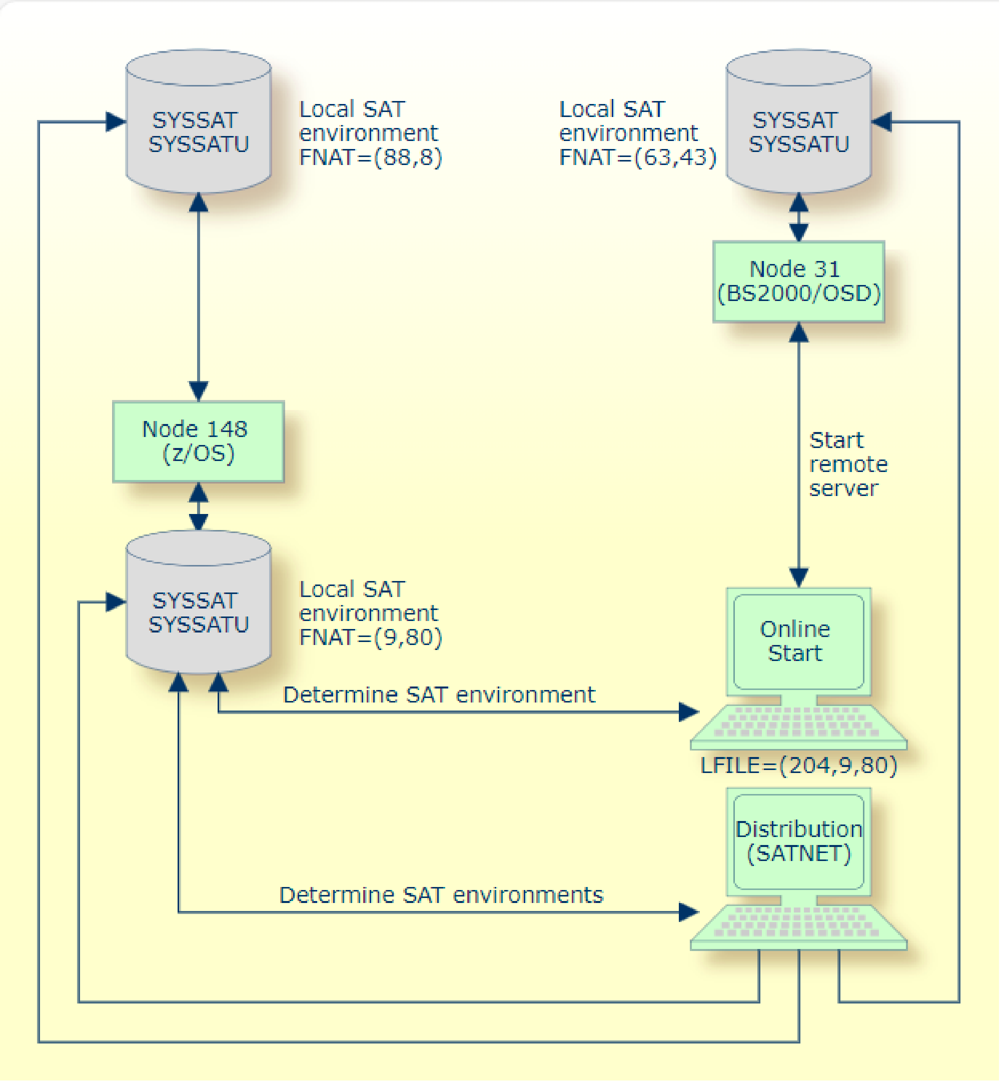
The entries in `SATDIR` are used to distribute your definitions into your System Automation Tools environments (with the `SATNET` program) and to determine the local System Automation Tools environment for a specific node when starting servers from online.

The following topics are covered below:

- System Automation Tools in Distributed Computing Environments

- SATDIR Syntax

System Automation Tools in Distributed Computing Environments



The above illustration shows the following scenario: A user has logged onto Natural, whose LFILE entry for ID=204 points to `FNAT=(9,80)`. The user's main text object `SP00148` and the text object `SATDIR` reside in the `SYSSATU` library of that FNAT system file. With this connection, the user can start System Automation Tools product servers online.

SATDIR Syntax

```
SATnnnnn SATDIR   SATSYSF=(sat-dbid,sat-fnr)
```

where:
SAT*nnnnn* is the Entire System Server node number,
sat-dbid is the database ID of the local FNAT,
sat-fnr is the file number of the local FNAT.

Example:

```
SAT00148   SATDIR   SATSYSF=(9,80)   /* Default settings for node 148.
```

This line reflects the scenario above.

11 Definitions for EntireX Broker Access - SATBKR

After the installation of Entire Operations or Entire Output Management, you have to create a Natural text object SATBKR in the library SYSSATU, as described in this section.

You have to customize the text object SATBKR to contain the required parameter definitions for System Automation Tools.

For each EntireX Broker which is to be accessed with Broker Security and/or SSL, you need one section of parameter definitions in SATBKR. Each section has the following format:

```
BROKER-ATTRIBUTES
  BROKER-ID=broker-id
  USERID=user-id
  CPW=ciphered-password
  SSL-TRUST-STORE=SSL-trust-store
```

For Brokers without Broker Security and without SSL, you do not need an entry in SATBKR.

The individual parameters are described below.

For example definitions, see the text object SATBKREX in the library SYSSATU, which can be used as a template for SYSBKR.

Note: No additional definitions are required in the invoking Entire Systems Management products. Broker Security and SSL support is fully transparent for their node definitions.

Creating SATBKR for the First Time

If you are installing System Automation Tools for the first time, create SATBKR as follows:

1. Invoke Natural and log on to the library SYSSATU.
2. Issue the command EDIT SATBKREX.
3. Issue the command SAVE SATBKR.

SATBKR Parameters

Parameter	Explanation
`BROKER-ID`	Corresponds to the parameter `BROKER-ID` specified in the EntireX Broker attribute file. For information on the maximum length of the `BROKER-ID` parameter, see the EntireX Broker documentation. *`broker-id`* is the name of the EntireX Broker under which the service is started, for example, `BRK`*`nnn`*. If Entire Operations or Entire Output Management is executed in a UNIX or a Windows environment, for *`broker-id`* the following syntax can be used: *`host`*`:`*`port`*`:TCP`. Example: `BROKER-ID=ibm1:29000:TCP` **Note:** A `SATBKR` definition will only be used by System Automation Tools if the `BROKER-ID` specified in `SATSRV` is the same as the one specified here.
`USER-ID`	Corresponds to the parameter `USER-ID` specified in the EntireX Broker attribute file. *`user-id`* is the user ID for the mainframe or the UNIX server.

`CPW`	For Broker Security:
	The password for the *`user-id`*, in ciphered format.
	To cipher a password, use the utility `nprpwc` (UNIX) or `nprpwc.exe` (Windows), which is delivered with Entire System Server for UNIX and Windows respectively. How to use:
	UNIX: Open a console (shell) window: `cd $NPRDIR/$NPRVERS/bin nprpwc`
	Windows: Open a DOS (command prompt) window:
	`cd %nprdir%\%nprvers%\bin`
	`nprpwc.exe`
	Note: After any password change, the ciphering must be repeated.

`SSL-TRUST-STORE`	Optional, for SSL communication: The SSL Trust Store, as described in the EntireX Broker documentation. SSL communication will be used only if this parameter is specified.

12 Definitions for Entire System Server - SATSRV

This section only applies to Entire System Server nodes on UNIX and Windows.

After the installation of Entire Operations or Entire Output Management, you have to create a Natural text object SATSRV in the library SYSSATU, as described in this section.

You have to customize the text object SATSRV to contain the required parameter definitions for System Automation Tools.

For each service that is to be accessed via EntireX Broker, you need one section of parameter definitions in SATSRV. The entries in SATSRV have the format:

```
node-name   SATSRV TYPE=ACI
                   BROKER-ID=broker-id
                   SERVER-CLASS=NPR
                   SERVER-NAME=server-name
                   SERVICE=service-name
                   USER-ID=user-id
                   WAIT-TIME=seconds
                   LOCALE-STRING=locale_string
```

The individual parameters are described below.

For example definitions, see the text object SATSRVEX in the library SYSSATU, which can be used as a template for SATSRV.

To access a service in local mode (without using EntireX Broker), certain parameter definitions are required. Copy the second section of the example in SATSRVEX into SATSRV and replace *service-name* with a name of your choice.

Creating SATSRV for the First Time

If you are installing System Automation Tools for the first time, create SATSRV as follows:

1. Invoke Natural and log on to the library SYSSATU.
2. Issue the command EDIT SATSRVEX.
3. Issue the command SAVE SATSRV.

SATSRV Parameters

Parameter	Explanation
node-name	It is recommended that the same name as the *server-name* be used. For Entire Operations, see note below.
TYPE	The type of communication: It must always be ACI.
BROKER-ID	Corresponds to the parameter BROKER-ID specified in the EntireX Broker attribute file. For information on the maximum length of the BROKER-ID parameter, see the EntireX Broker documentation. *broker-id* is the name of the EntireX Broker under which the service is started, for example, BRK034. If Entire Operations or Entire Output Management is executed in a UNIX or a Windows environment, for *broker-id* the following syntax can be used: *host*:*port*:TCP. Example: BROKER-ID=ibm1:3800:TCP **Note:** A SATBKR definition will only be used by System Automation Tools if the BROKER-ID specified in SATBKR is the same as the one specified here.
SERVER-CLASS	Corresponds to the parameter CLASS specified in the EntireX Broker attribute file.
SERVER-NAME	Corresponds to the parameter SERVER in the SDPA structure. *server-name* is the name of a mainframe or a UNIX server. This is the host name of the machine on which the Broker Service is to be executed. The 'Server=' specification in the npr.ini file must be identical to the *server-name* specified here. For Entire Operations, see note below.
SERVICE	Corresponds to the parameter SERVICE specified in the EntireX Broker attribute file. *service-name* is the name of the EntireX Broker Service.
USER-ID	This parameter is obsolete. For compatibility reasons, however, it can remain part of SATSRV; that is, you need not remove it.
WAIT-TIME	Corresponds to the parameter WAIT specified in the EntireX Broker attribute file. *seconds* is the 3-character wait time in seconds, for example, 60S.
LOCALE-STRING	Corresponds to the parameter LOCALE_STRING in the SDPA structure. The parameter is important for character-set translation and conversion.

Note for Entire Operations:

The `node-name` and the `server-name` must correspond to the node names specified in the Entire Operations node table; see *Definition of Nodes* in the *Entire Operations Administration* documentation.

Also,`node-name` and `service-name` must correspond to a section name within the `npr.ini` file on the target system. We recommend that the same identifiers be used for node names and service names.

13 Starting a Server

Different methods are supported for starting a server of an Entire Systems Management product.

ONLINE-Start

On z/OS, the start of a server of any Entire Systems Management product can be performed online:

- Entire Operations
- Entire Output Management
- Entire Event Management

Proceed as follows:

1. Use an online Natural with the following specifications:
   - `FNAT` must contain the `SYSSAT` library (as installed in Step 2).
   - `LFILE 204` must point to your local `SYSSAT` environment in order to find main text object `SP00148` in `SYSSATU` (see **Step 6** and **Step 7** of the *Installation Procedure*).
2. Log on to the appropriate product library, for which you want to start the server(s).
3. Invoke the product-specific start command (see the documentation for the product itself).
4. This start command reads the `SATSTART` parameter block of the appropriate product in `SP00148` and invokes the server initialization program.
5. You will be informed online about the success of the operation.

AUTO-START

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 4** and **Step 5**):

   This must be a subtask-Natural, because it runs in the address space of the Entire System Server.
2. To activate this process, adapt the startup parameters of Entire System Server:
   - Specify the name of the Natural module which should be given control.
   - Specify the `LOGON` commands to invoke the program `SATSTART` in the library `SYSSAT`.

   If Natural Security is installed, specify the following parameters and supply the appropriate parameter values if required:

```
NATSHARE=nucleus-name
NATNUMSUB=subtask-maximum
NATMOD=subtask-module
STRTNTP1=STACK=(LOGON SYSSAT,nsc-user,nsc-pswd;
STRTNTP2=SATSTART;FIN),AUTO=OFF
```

where:

nucleus-name is the name of Natural shared nucleus if used.

subtask-maximum is the maximum number of subtasks (recommended: 20).

subtask-module is the name of Natural subtask module as linked in Step 4.

nsc-user is the user ID required to log on to Natural Security.

nsc-pswd is the password required for logging on to Natural Security.

If Natural Security is not installed, specify the following parameters:

```
STRTNTP1=STACK=(LOGON SYSSAT;SATSTART;FIN),AUTO=OFF
```

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 text object.
4. For each `SATSTART` instruction defined in the `SP00148` text object, `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 object specified with the `NATSKEL` parameter (library is `SYSSATU`). The jobname of the server task is created automatically.

   For more information on the above, see ***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:

```
productSAT
```

where *product* denotes the 3-letter code of the respective product, for example, `NOPSAT`.

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.

The following topics are covered below:

- Starting Servers with TYPE=SUBTASK
- Starting Servers with TYPE=BATCH

Starting Servers with TYPE=SUBTASK

For each `SATSTART` instruction, in the address space of Entire System Server, a subtask is started which initiates the server start. The subtask name is built as follows:

```
pppSTAdddddfffff
```

where:

ppp = product code

ddddd = DBID as specified in the `SERVSYSF` parameter

fffff = FNR

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

1. Adapt either the `NSBTSKIS` text object (for ISP format) or the `NSBTSKSD` text object (for SDF format) in the `SAT`*vrs* source library.

   Note: The `ADALNK` parameter file is optionally supported. To use this function, you have to change the text object `NSBTSKIS` or `NSBTSKSD` correspondingly. Further information is available in the current *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 user ID as specified in the `ESYUSER` parameter. The job names of these batch jobs are built as follows:

```
pppSTnnn
```

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:

```
pppnnnrr
```

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 the `SYSSAT` library 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
```

RPC Servers

If an Entire Systems Management PC product is installed, which is connected to a mainframe monitor (like Output Management GUI Client) it may be useful to start several RPC servers in parallel to balance data traffic. This could be done by starting several Natural RPC servers as batch jobs, but administration is easier, if start and stop is controlled by System Automation Tools.

For this purpose, there is the following:

- A new product code `RPC` for System Automation Tools to start and stop Natural RPC servers like monitors started for Entire Output Management or Entire Operation. This means that RPC servers will come up when Entire System Server is started and be ready for use until Entire System Server stops.
- A new Entire System Server console command `SHUTDOWN RPC` to end all RPC servers started by `SATSTART`.

- An extra task "WATCHDOG": since no Natural program is executed in a Natural RPC server, we need an extra task that controls start and stop procedures. This task issues the control commands to shutdown RPC servers, triggered by Entire System Server (at shutdown time of Entire System Server or if a `SHUTDOWN RPC` command occurs).

14 Event Store

This section describes the System Automation Tools event store, and covers the following topics:

General Information on Event Store

The event store of System Automation Tools is available in conjunction with Entire Operations.

The event store can be used to collect events which may occur during a network run. The collected events can then be transferred to an external event management system for further processing and evaluation. This may be useful to identify error situations or monitor the network run.

You define milestones at which event data are written. These event data are stored in a separate database file. From that file, the event data can be transferred - via a Natural RPC server call - to an external RPC server.

The transfer procedure uses a monitor program which ensures that the events are transferred in chronological order. Information about the success/failure of the transfer is returned from the receiving RPC server to the monitor and stored in the event store file. If an event was not transferred successfully, its transfer will be repeated.

If a network has been started using the Entire Operations business API (B-API), correlation information is written. This information can be used by the external event management system to ascertain which event data are related to which network. Moreover, the event data written for milestones set for a network contain a unique network identification.

Event Store Milestones

The milestones at which events are collected can be:

- jobs defined as milestone jobs,
- automatic network milestones,
- milestones for global messages,
- user-defined milestones.

Jobs Defined as Milestones Jobs

You can define each job in a network as a milestone job. This is done in the Entire Operations job master definition.

A milestone job can be:

- a network start job (the first relevant job in a network),
- a network end job (the last relevant job in a network),
- any other job.

In each network, there should be only one start job and one end job.

The job-end processing of an "other" milestone job generates a corresponding "OK" or "not OK" milestone.

Automatic Network Milestones

For a network, event information for the following events is collected automatically:

- network activation,
- network release,
- network deactivation.

For an activated network, event information for the following activated jobs is also collected automatically: the job that was started first and the job that ran last.

Milestones for Global Messages

If Entire Operations global messages are activated for a network for which events are written, the globals messages will also be treated as an event. This allows you to store additional error situation information.

The events for which global messages are written are defined in Entire Operations on the **Global Messages for Events** screen (see *Entire Operations* documentation).

User-Defined Milestones

The Entire Operations application programming interface (API) `NOPUMI1N` allows you to define your own milestones. It is described in the *Entire Operations User's Guide*.

Event Store File

The file in which the events collected by the event store are to be stored has to be specified as file `84` with the Natural parameter `LFILE=(84,`*`dbid`*`,`*`fnr`*`)` has to be specified, *`dbid`* and *`fnr`* being the database ID and file number of the event store file.

The database ID and file number of the event store file are displayed in the Entire Operations monitor defaults.

Event Store Administration

This section describes the programs used to control the event store:

- Deleting Old Events - SAECLE-P
- Sending Commands to the Event Store Monitor - SAECMD-P
- Starting the Event Store Monitor - SAEM--0P
- Checking the Status of the Event Store - SAEYPI0N

All these programs are available in the library `SYSSAT`.

Deleting Old Events - SAECLE-P

To delete events from the event store file, you use the program `SAECLE-P`, which is invoked with the following parameters:

```
SAECLE-P P-RETENTION-DAYS P-UNPUBLISHED-ALSO
```

Example:

```
SAECLE-P 20 A
```

Parameter	Format/Length	Input/Output	Explanation
`P-RETENTION-DAYS`	`N29`	input	*`nn`* = Delete all successfully transferred events which are older than *`nnn`* days.
`P-UNPUBLISHED-ALSO`	`A1`	input	`A` = Also delete events which have not been transferred. (optional).

Note: If you wish events to be deleted automatically at regular intervals, you can create a specific Entire Operations network which performs an appropriate task.

Sending Commands to the Event Store Monitor - SAECMD-P

The event store monitor checks at regular intervals for commands to be processed. To send a command to the monitor, you use the program `SAECMD-P`, which is invoked as follows:

```
SAECMD-P command
```

Parameter	Format/Length	Input/Output	Explanation
`command`	`A8`	input	Possible values: ■ `STOP` = Stop the event store monitor. ■ `FREE` = Clear all sent commands. This may be necessary, if the monitor has stopped without a `STOP` command having been sent.

Starting the Event Store Monitor - SAEM--0P

The event store monitor is a program which checks at regular intervals if new events have been stored in the event store file. It transfers the events from the event store file to a RPC server program which transfers them to the external event management system.

The event store monitor is started with the following command:

```
SAEM--0P P-NAME-UM-SUBP P-SEL-CORRELATION-SUBSYS P-RETURN-EVENTS P-WAIT-SEC ↩
P-BROKER-ID P-SERVICE
```

Example:

```
SAEM--0P SAEGWSON * 3 10 integServer:1971 ESMGWSRV
```

Parameter	Format/Length	Input/Output	Explanation
P-NAME-UM-SUBP	A8	input	The name of the client stub subprogram generated for the RPC service. The stub subprogram SAEGWSON provided in the library SYSSAT should be used.
P-SEL-CORRELATION-SUBSYS	A32	input	Determines whether events are transferred for: ■ ' ' (blank) = all scheduled networks; ■ '*' = all subsystems; ■ '*subsystem-id*' = this correlation subsystem.
P-RETURN-EVENTS	I4	input	The number of events to be transferred with each call.
P-WAIT-SEC	I4	input	The time interval between two successful transfers (in seconds).
P-BROKER-ID	A70	input	The name of the broker or integration server, and the port.
P-SERVICE	A32	input	The name of the service to be called. This must be ESMGWSRV:

The invoked RPC server program has to implement the following interface (IDL):

```
/* Generated by Software GmbH, IDL Extractor for IS
/* from IS package COM_ESM
/* Optimized for usage with Natural Wrapper.
Library 'COM_ESM' Is
 Program 'SAEGWSRV' Is /* com.esm.listenerService:SAEGWSRV
  DEFINE DATA PARAMETER
  1 ESM-EVENT-STORE-ARR (/V) inout
   2 ES-EVENT-ID (AV)
   2 ES-SUBSYS (AV)
   2 ES-CORRELATION-SUBSYS (AV)
   2 ES-CORRELATION-ID (AV)
   2 ES-CORRELATION-TYPE (AV)
   2 ES-ESM-CORRELATION-ID (AV)
   2 ES-CREATION-TIMESTAMP (AV)
   2 ES-PUBLISH-TIMESTAMP (AV)
   2 ES-PUBLISH-STATUS (AV)
   2 ES-PUBLISH-RCCODE (AV)
   2 ES-PUBLISH-RCTEXT (AV
   2 ES-EVENT-TYPE (AV)
   2 ES-EVENT-VERSION (AV)
   2 ES-EVENT-DATA (AV)
  1 UM-SERVER-ERR-BLOCK inout
   2 UM-RC (AV)
   2 UM-RC-TEXT (AV)
  END-DEFINE
```

The parameters are explained below.

<table>
<tr><th>Parameter</th><th>Format/Length</th><th>Input/Output</th><th colspan="2">Explanation</th></tr>
<tr><td colspan="2">ESM-EVENT-STORE-ARR</td><td>input/output</td><td colspan="2"></td></tr>
<tr><td>ES-EVENT-ID</td><td>A36</td><td>input</td><td colspan="2">The unique event identifiier.</td></tr>
<tr><td>ES-SUBSYS</td><td>A32</td><td>input</td><td colspan="2">The system for which the event has been generated.
For example: NOP = Entire Operations.</td></tr>
<tr><td>ES-CORRELATION-SUBSYS</td><td>A32</td><td>input</td><td colspan="2">The correlation subsystem. This is defined by the external event management system to determine different messaging queues.</td></tr>
<tr><td>ES-CORRELATION-ID</td><td>A36</td><td>input</td><td colspan="2">The correlation ID which identifies the run or job network. This unique ID is supplied by the external event management system.</td></tr>
<tr><td>ES-CORRELATION-TYPE</td><td>A32</td><td>input</td><td colspan="2">The correlation type. This is defined in the external event management system as an additional identifier.</td></tr>
<tr><td>ES-ESM-CORRELATION-ID</td><td>A60</td><td>input</td><td colspan="2">The unique correlation ID defined by the event store.</td></tr>
<tr><td>ES-CREATION-TIMESTAMP</td><td>A23</td><td>input</td><td colspan="2">The timestamp when the event ocurred (in the format YYYYMMDDHHIISSTTT).</td></tr>
<tr><td>ES-PUBLISH-TIMESTAMP</td><td>A23</td><td>output</td><td colspan="2">The timestamp when the event was transferred (in the format YYYYMMDDHHIISSTTT).</td></tr>
<tr><td>ES-PUBLISH-STATUS</td><td>A1</td><td>input/output</td><td colspan="2">The transfer status: ' ' (blank) = transferred; 'N' = not transferred.</td></tr>
<tr><td>ES-PUBLISH-RCCODE</td><td>I4</td><td>input/output</td><td colspan="2">The transfer response code: 0 = transferred successfully; any other value = error during transfer.</td></tr>
<tr><td>ES-PUBLISH-RCTEXT</td><td>A DYNAMIC</td><td>output</td><td colspan="2">Description of transfer error.</td></tr>
<tr><td rowspan="8">ES-EVENT-TYPE</td><td rowspan="8">A32</td><td rowspan="8">input</td><td colspan="2">Entire Operations event type and trigger:</td></tr>
<tr><td>ON_API<i>name</i></td><td>User API with a name that begins with ON_API.</td></tr>
<tr><td>ON_ACTIVATE</td><td>Network activation.</td></tr>
<tr><td>ON_RELEASE</td><td>Network release.</td></tr>
<tr><td>ON_DEACTIVATE</td><td>Network deactivation.</td></tr>
<tr><td>ON_1ST_JOB</td><td>First job.</td></tr>
<tr><td>ON_LAST_JOB</td><td>Last job.</td></tr>
<tr><td>ON_MILESTONE_START</td><td>Job with milestone has been started.</td></tr>
</table>

Parameter	Format/Length	Input/Output	Explanation	
			`ON_MILESTONE_END`	Job with milestone has ended.
			`ON_MILESTONE_OK`	Job with milestone ended with `OK`.
			`ON_MILESTONE_NOT_OK`	Job with milestone ended with `NOT OK`.
			`ON_GLOBAL_MESSAGE`	For activated global messages.
`ES-EVENT-VERSION`	`A5`	input	The format and version of the data contained in `ES-EVENT-DATA`: `PT.01` = plain text, unformatted; `SX.01` = very simple XML. See *Event Data Format* below.	
`ES-EVENT-DATA`	`A DYNAMIC`	input	Additional event data of the above format.	
`UM-SERVER-ERR-BLOCK`		output		
`UM-RC`	`I4`	output	The error code if the transfer failed completely. `0` = transfer successful.	
`UM-RC-TEXT`	`A DYNAMIC`	output	The error text(s) of the error(s) occurred.	

Event Data Format

Format	Explanation
Plain text, unformatted	Code translation of the event data is performed by EntireX during the transfer. For the event data, use only structured formats which to not rely on internal encoding information. Do not use structured data based on line feeds or or carriage returns, as these may be converted to blanks by the EntireX default settings.
Very simple XML	To avoid code-page translation problems as outlined for plain text above, you can use a very simple subset of XML: Do not use XML headers, XML name spaces, XML attributes, or any XML element names which contain lower-case characters. Use only single-character encoding, based on the current Natural alphanumeric definition. For special-character encoding, use only `&amp;` for "&", `&lt;` for "<", and `&gt;` for ">".

Templates to Start the Monitor

On mainframes, the JCL template `SESKLMVS` in the library `SYSSAT` can be adapted and used to start the event store monitor. On UNIX, the template `satserv.bsh.tpl` is available for the same purpose.

Checking the Status of the Event Store - SAEYPI0N

To check the status of the event store and obtain various items of status information, you use the application programming interface SAEYPION, which is invoked as follows:

```
CALLNAT 'SAEYPION'
  P-STATUS P-STARTED
  P-BLOCK-SIZE P-WAIT-INTERVAL P-RETENTION-PERIOD
  P-UM-RC P-UM-RC-TEXT P-UM-RC-LONG
```

It returns the following information:

Parameter	Format/Length	Input/Output	Information Returned

P-STATUS	A DYNAMIC	output	Status of the event store monitor:
			A = monitor active.
			C = monitor cancelled (stopped).
			N = event store not found.

P-STARTED	A DYNAMIC	output	Last start time.
P-BLOCK-SIZE	I4	output	Current block size of transferred events (number of events transferred per call).
P-WAIT-INTERVAL	I4	output	Current minimum waiting time (in seconds) between two transfer calls.
P-RETENTION-PERIOD	I4	output	Retention period after which transferred events are removed from the event store file.
P-UM-RC	I4	output	Error number (corresponds to the Natural error number; for example: `3061: Check LFILE settings`).
P-UM-RC-TEXT	A DYNAMIC	output	Error short text.
P-UM-RC-LONG	A DYNAMIC	output	Error long text.

15 Messages

To display a System Automation Tools message in Entire Operations, you enter the Entire Operations direct command:

```
HELP MSG SATnnnn
```

where *nnnn* is the error number. The message will be displayed in the language in which Entire Operations is being used.

To display a System Automation Tools message in Natural, you enter the Natural system command:

```
HELP U nnnn SYSSAT
```

where *nnnn* is the error number. The message will be displayed in the language in which Natural is being used.