Sun N1 Advanced Architecture for SAP Solutions 5.2.1 User's Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819–5458–10
May 2006

060508@14986

Contents

Preface

The *Sun N1™ Advanced Architecture for SAP Solutions 5.2.1 User's Guide* explains how to use the Sun N1 Advanced Architecture for SAP Solutions software.

Who Should Use This Book

This book is intended for users of the Sun N1 Advanced Architecture for SAP Solutions software.

Before You Read This Book

If you are not already familiar with using the Sun N1 Service Provisioning System software, read the following books:

- *Sun N1 Advanced Architecture for SAP Solutions 5.2.1 Installation Guide*
- *Sun N1 Service Provisioning System User's Guide for SAP Plug-In 1.0*
- *Sun N1 Service Provisioning System User's Guide for OS Provisioning Plug-In 3.1*
- *Sun N1 Service Provisioning System 5.2 Plan and Component Developer's Guide*

How This Book Is Organized

Chapter 1 contains an overview of Sun N1 Advanced Architecture for SAP Solutions.

Chapter 2 describes the configuration and usage of the N1 AA base components.

Chapter 3 describes the setup and usage of the N1 AA Analyzer module.

Chapter 4 describes the setup and usage of the N1 AA Builder module.

Chapter 5 describes the setup and usage of the N1 AA Deployer module.

Chapter 6 describes the Command Line Interface (CLI) commands.

Appendix A contains information of the N1 AA Analyzer Server Group Utilization and MDX queries.

Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.sun.com/training/`)

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename.*

TABLE P–1 Typographic Conventions *(Continued)*

Typeface	Meaning	Example
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX® system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

Shell	Prompt
C shell	`machine_name%`
C shell for superuser	`machine_name#`
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell for superuser	`#`

◆ ◆ ◆ CHAPTER 1

N1 AA Overview

With Sun N1 Advanced Architecture for SAP Solutions, IT operators can deploy, provision, manage, and change pools of resources and application services, and analyze the environment, all from a single Web browser interface. Sun N1 Advanced Architecture for SAP Solutions delivers more business flexibility through virtualization, application and resource provisioning, monitoring, and central management.

For a more technical view on the architecture, see the *Sun N1 Advanced Architecture for SAP Solutions 5.2.1 Installation Guide*.

N1 AA Modules

Sun N1 Advanced Architecture for SAP Solutions consists of the central core module N1 AA Manager and the three additional functional modules Analyzer, Builder, and Deployer.

- N1 AA Manager

  The N1 AA Manager provides the base components, or the platform, for general customization, maintenance, and administration. The specific *functional* features of the solution are provided by the Analyzer, Builder, or Deployer modules.

- N1 AA Analyzer

  In a systems environment, information on the current utilization of resources is the starting point for making smarter decisions. The N1 Advanced Architecture Analyzer continually monitors, collects, and displays CPU and memory utilization data on all of the servers and applications in the environment. This includes applications other than SAP. Once data is collected, the Analyzer provides reports on capacity planing and graphical illustrations of resource consumptions.

- N1 AA Builder

  The N1 AA Builder is designed for OS provisioning to quickly provision servers into an SAP environment from a bare metal state. The pre–built models within the Builder enable IT operators to build, configure, and update multi-tier application service models for many data center applications. These include models for the operating systems, patches, and client software

such as backup software or the N1 AA clients, all from a single GUI. The models are stored and centrally administered. The models can be configured for different environments.

- N1 AA Deployer

  The Deployer is the controller for your application services. The Deployer enables you to start, stop, and relocate applications in a virtualized environment. This includes applications, that are driven in a Sun Cluster or SAP Adaptive Computing (SAP AC) environment. For SAP systems, the Deployer enables you to install new SAP Application Instances or remove existing ones.

For more information, see the *Sun N1 Advanced Architecture for SAP Solutions 5.2.1 Installation Guide* for a technical view of the architecture.

◆ ◆ ◆ CHAPTER 2

N1 AA Manager

This chapter describes the configuration and use of N1 AA base components. These base components are functions and features with an overall relationship across all of the N1 AA modules.

User Management

Login accounts for the N1 AA Manager are created and managed at the OS-level. For more information, see *Sun N1 Advanced Architecture for SAP Solutions 5.2.1 Installation Guide*.

Customizing

General Customizing

Purpose	General customizing for N1 AA
Menu	Administration ⇒ Customizing
Tasks for an initial basic setup	Mandatory

FIGURE 2–1 General Customizing

General customizing for N1 AA is done here. The following parameters are supported:

TABLE 2–1 General Customizing Parameters

Parameter	Description	N1 AA Module	Value
`none.general.auth.analyzer`	Determines if the Analyzer menu is available in the GUI or not	Mandatory for: Analyzer	Insert this parameter with any value to make the menu available
`none.general.auth.builder`	Determines if the Builder menu is available in the GUI or not	Mandatory for: Builder	Insert this parameter with any value to make the menu available
`none.general.auth.deployment`	Determines if the Deployer menu is available in the GUI or not	Mandatory for: Deployer	Insert this parameter with any value to make the menu available

TABLE 2–1 General Customizing Parameters *(Continued)*

Parameter	Description	N1 AA Module	Value
`none.general.n1aa.communication`	Defines the communication from the N1 AA Server to the N1 AA Clients. The Analyzer uses simple OS commands on the N1 AA Clients to import the Performance Collector data: ■ Localhost: N1 AA Server ■ Remote host: All N1 AA Clients ■ OS User: noaccess ■ OS Commands: `tail`, `cat`, and `logadm` Configure this parameter with the prefix to use in front of the fixed coded part. See examples. **Note –** You can add an additional parameter *servergroup*`.general.n1aa.communication`. For example, if there is a server group `DEV` configured, than you can overwrite the value customized by parameter `none.general.n1aa.communication` specifically in the server group `DEV` by adding the parameter `dev.general.n1aa.communication`. This overwrites the parameter `none.general.n1aa.communication` within this dedicated server group. You could configure the communication type server group dependently. This parameter must be in line with the configured communication method OS level. This is normally done during the installation process of N1 AA. For more information see the *Sun N1 Advanced Architecture for SAP Solutions 5.2.1 Installation Guide*.	Mandatory for: Analyzer	Examples: Value = /opt/SUNWn1aa/aasap/bin/aasap Command, that is used within N1 AA: /.../aasap *remotehost command* Value = rsh Command that is used within N1 AA: rsh *remotehost command* Value = ssh Command that is used within N1 AA: ssh *remotehost command* Value = ssh `-p` 5122 Command that is used within N1 AA: ssh `-p` 5122 *remotehost command*

TABLE 2–1 General Customizing Parameters *(Continued)*

Parameter	Description	N1 AA Module	Value
`none.general.n1aa.logfile`	Defines the location of the N1 AA logfile. **Note –** Make sure that the OS user `noaccess` has the rights to create the file in the target directory or create an empty log file manually. If you create the log file manually, make sure that the OS user `noaccess` has the rights to write to the file.	Mandatory for all modules	Example: `/var/opt/SUNWn1aa/n1aa.log`
`none.general.n1aa.loglevel`	Defines the log level to be used.	Mandatory for all modules	Available levels are: DEBUG, INFO, SUCCESS, WARNING, ERROR, FATAL, PANIC
`none.general.sps.installpath`	Installation path to the SPS Java API **Note –** N1 AA has to be restarted for changes on `none.general.sps.installpath` to take effect.	Mandatory for: Builder Deployer	Example: `/opt/SUNWn1sps/N1_Service_Provisioning_S`
`none.general.sps.user`	Specifies the user that N1 AA uses to log in to SPS.	Mandatory for: Builder Deployer	Example: spsusr
`none.general.sps.password`	Specifies the password, that N1 AA uses to log in to SPS.	Mandatory for: Builder Deployer	Example: spspw

Server Groups

Purpose	Definition of Server Groups
Menu	Administration ⇒ Customizing
Tasks for an initial basic setup	Mandatory

FIGURE 2–2 Server Groups

Server Groups, groups of physical hosts, allow you to split your landscape into administrative units. This splitting is reflected in the GUI. You always work inside a Server Group. Your actions are limited to the hosts belonging to the server group.

Define Server Groups

- Name – Short name of the Server Group
- Description – Description of the Server Group

Note – Be careful when deleting a server group. You lose all physical hosts including their settings and referenced data. For example, N1 AA Analyzer cube data. For more information, see Chapter 3. In order to keep these physical hosts, assign the hosts to another server group before deletion.

Physical Hosts

Purpose	Define hosts of the N1 AA landscape and assign them to Server Groups.
Menu	Administration ⇒ Customizing
Tasks for an initial basic setup	Mandatory

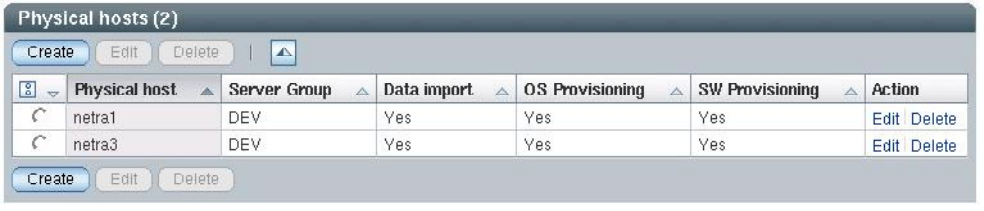

FIGURE 2–3 Physical Hosts

Maintain all physical hosts that are part of your N1 AA landscape.

- Physical host: Hostname (IP name) of the server
- Server Group: Assign the host to an existing group

- Date import (yes or no)
  - yes (default): The performance data of this host is regularly imported into the database by the import job. For more information, see "Data Imports" on page 21. This is the normal case.
  - no: This host's data is not imported. The import job ignores this host. Use this behavior in special situations. For example, if you know that the host is unavailable for a long period.
- OS provisioning (yes or no)
  - yes (default): This host is available in the N1 AA Builder as a target host for OS provisioning.
  - no: This host is locked for OS provisioning.
- SW provisioning (yes or no)
  - yes (default): This host is available in the N1 AA Builder as a target host for SW provisioning.
  - no: This host is locked for SW provisioning.

Caution – Deleting a physical host erases these settings and all data related to this entry. For example, N1 AA Analyzer cube data. For more information, see Chapter 3.

Attention: For N1 AA Builder - OS Provisioning only

Using OS Provisioning in N1 AA, the selected target hostname will automatically be extended by the suffix –`target`. This means that the resulting installation target is *hostname*–`target` . This is in line with the default settings within the N1 SPS OSP Plug–in. Do not modify these default settings within SPS. Otherwise, OS Provisioning does not work with the N1AA Builder.

For more information on how to manage hosts in the N1 SPS environment, see *Sun N1 Service Provisioning System 5.2 System Administration Guide* and *Sun N1 Service Provisioning System User's Guide for OS Provisioning Plug-In 3.1*.

Scheduler

This is a general description of the job scheduler. The Chapter 3, Chapter 4, and Chapter 5 describe the necessary jobs needed for the operative business.

Purpose	Define jobs scheduled by N1 AA and view their logs
Menu	Administration ⇒ Scheduling
Tasks for an initial basic setup	None

Based on jobs and the scheduler, N1 AA can handle asynchronous and periodically scheduled task executions in background threads. Jobs will be automatically scheduled in the background (for example if you start or stop a resource in the N1 AA Deployer), but you can create jobs manually too.

- Name: Name of the job.

- Class: Fully-qualified class name of an N1 AA scheduling job, which is a descendant of class `com.sun.web.admin.n1aa.common.SchedulingJob`.
- Parameters: Parameters of the job. Spaces to separate parameters and quotation marks if a space is contained in a parameter. Syntax is analogous to console line arguments.
- Persistence
  - Persistent: Job definition and its log is saved permanently in the database. All data is available after a restart of N1 AA.
  - Transient: Job definition and its log is lost after a restart of N1 AA.
- Start: Date and time this job is should be executed for the first time.
- Interval: Restart period in seconds. A value of 86400 will define a daily interval. An empty field means that the job is immediately re-executed after it is finished.
- Repeats: Number of restarts that will appear on this job. A value of 1 specifies this job to be executed once, 2 for twice and so on. Empty field means an infinite repetition.

Each job collects messages into its own log that can be viewed. You find scheduling events, for example, `Job started` or `Job finished`, job info messages or errors along with a timestamp and a log level.

The default values are set to show only the most recent log lines at log level INFO. You can change these settings, For example, to see the complete log or ERROR messages only.

◆ ◆ ◆ CHAPTER 3

N1 AA Analyzer

This chapter describes the setup and usage of the N1 AA Analyzer module.

Overview

The Analyzer is a tool for capacity monitoring and planing. The concepts are:

- Data recording

  The resource consumption in the landscape is recorded.

Where is data recorded ?	Locally on every server of the landscape where the Performance Collector daemon is installed and started. For more information about the Performance Collector, see the *Sun N1 Advanced Architecture for SAP Solutions 5.2.1 Installation Guide*.
What is recorded ?	Resource consumption, CPU and Memory usage, of every running SRM project. For more information, see the *Sun N1 Advanced Architecture for SAP Solutions 5.2.1 Installation Guide*.

- Data import

  A job is scheduled for daily execution in the N1 AA Manager. The resource consumption data is imported from every server in the landscape into the database.

- Data analysis

  The N1 AA Analyzer provides an engine for the data analysis of the data. This analysis is based on Multi-Dimensional Expressions (MDX) Queries. There are predefined queries delivered along with N1 AA. You can also write your own queries.

Basic Setup

Analyzer Customizing

Purpose	Customizing for N1 AA Analyzer
Menu	Administration ⇒ Analyzer ⇒ Customizing
Tasks for an initial basic setup	All changes are optional

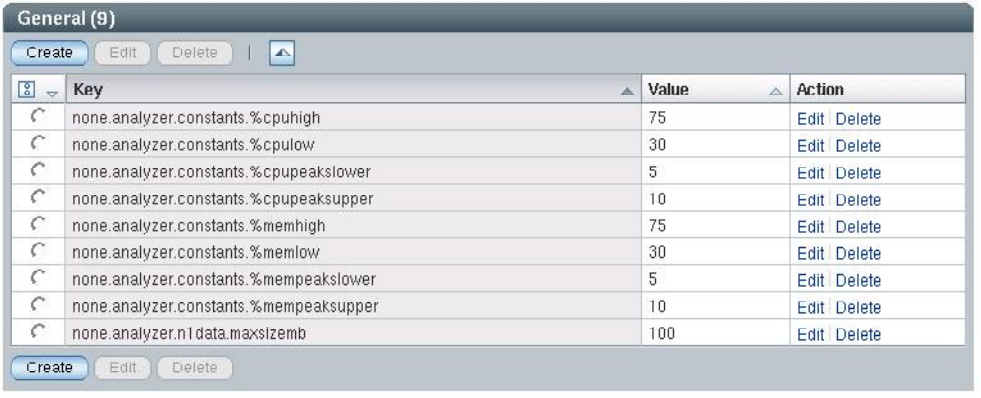

FIGURE 3–1 General Customizing for Analyzer

Note –

- The entries with the naming convention *servergroup*`.analyzer.constants.`*constantname* are passed to the MDX Queries. Where *servergroup* is a placeholder for `none` or an existing server group, and *constantname* is a placeholder for any name you choose.
- The predefined constants delivered along with N1 AA are described in "N1 AA Analyzer: Server Group Utilization" on page 73.
- You may add entries here, if you need them for your own MDX Queries. For more information, see "Views" on page 26.

N1 AA Analyzer specific customizing is done here. The following parameters are supported:

TABLE 3–1 Analyzer Parameters

Parameter	Description	Example value
`none.analyzer.n1data.maxsizemb`	Maximum size of the local Performance Collector file in MB. This file, `/var/opt/SUNWn1aa/n1data`, exists on every N1 AA client. The data import job reorganizes this file if it exceeds the customized size (log rotation). The parameter `none.analyzer.n1data.maxsizemb` is server group spanning valid. This means that the value is set for all server groups.If the parameter *servergroup*`.analyzer.n1data.maxsizemb` exists, it overwrites the parameter `none...` within this specific server group.	5

Data Imports

Purpose	Import resource consumption data collected by the Performance Collector on the N1 AA Clients into the database
Menu	Administration ⇒ Scheduling
Tasks for an initial basic setup	Optional - but recommended

Even if you can trigger a data import into the database manually, see "Maintenance" on page 33, it is easier to schedule a regular data import job in the background. To do this, create an N1 AA scheduling job by using the following properties:

- Name = `AnalyseDataImportJob`

  Note – Use this exact name.

- Class = `com.sun.web.admin.n1aa.analyzer.AnalyseDataImportJob`
- Persistence = Persistent
- Start = Enter the date and time this job is to be first executed.
- Interval = Recommended value: 86400 (daily import).

The data import is executed for each physical host that has the Data Import option enabled, regardless of the server group. For more information, see "Customizing" on page 11. Note that

within this execution, existing repository information is assigned properly, while new repository information is automatically created using default properties which you have to maintain. Refer to the following table:

TABLE 3–2 Repository Information

New repository information	Behavior	Default properties
New CPU type name or speed	A new CPU type is being created. See "CPU Types" on page 22.	Normfactor = 1.0
New SRM project	A new component is being created. See "Applications" on page 23 and "Components" on page 24.	Name = SRM project Application = None

Reorganizations

Purpose	Reorganization of resource consumption data
Menu	Administration ⇒ Scheduling
Tasks for an initial basic setup	Optional - but recommended

Having too much informations in the N1 AA Analyzer cube (historical resource consumption data) slows down response times. Therefore, old data can be moved to an archive table contained in the N1 AA database, see "Maintenance" on page 33. It is your choice if this should be done periodically without user interaction. All you have to do is create an N1 AA scheduling job using the following properties:

- Name = ArchiveUnarchiveJob
- Class = `com.sun.web.admin.n1aa.analyzer.ArchiveUnarchiveJob`
- Parameter = a "timestamp '-infinity'" "timestamp without time zone 'now' - interval '12 month'"

  Note – Use this exact string beginning with the character `a`. This means, that all data that is older than 12 months is archived. You may change the period of 12 months to a shorter value, for example 6 months, if your data growth is very high.

- Persistence = Persistent
- Start = Enter the date and time this job is to be first executed
- Interval = Recommend 86400 (daily reorganization)

CPU Types

Purpose	Maintain CPU normalization factors.

Menu	Administration ⇒ Analyzer ⇒ Customizing
Tasks for an initial basic setup	Mandatory - but can be done or changed at any time later on.

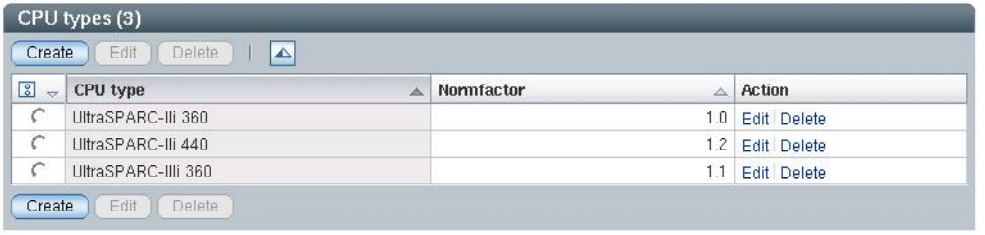

FIGURE 3–2 CPU Types

The CPU types are automatically recognized in your landscape and corresponding entries are automatically created for each new CPU type as performance data is imported. Normally, there is no need to create or delete an entry manually. However, it is important to maintain the proper CPU normalization factor after a new entry was created automatically.

- CPU type: Type of the CPU as recognized and applied by the N1 AA Analyzer.
- Normfactor: New entries created automatically by the N1 AA Analyzer data import job have the default value of 1.0. Adjust this value to reflect the relative CPU throughput as compared to the existing CPUs in your landscape.

The outcome of normalization is that N1 AA stores all measured CPU utilization. Based on the Normfactor of the CPU type, N1 AA calculates and also stores normalized CPU seconds. In the simplest case, this is consumed seconds * Normfactor. In this way, MDX queries can be defined based on CPU seconds or normalized CPU seconds. See the predefined MDX Queries as an example of how to use either CPU seconds or normalized CPU seconds.

Note – Exercise care when deleting CPU types. You lose all related resource consumption data that has been collected by the Performance Collector on the N1 AA Clients and imported into the database.

Applications

Purpose	Maintain applications in your N1 AA Analyzer landscape
Menu	Administration ⇒ Analyzer ⇒ Customizing
Tasks for an initial basic setup	Mandatory - but can be done or changed at any time

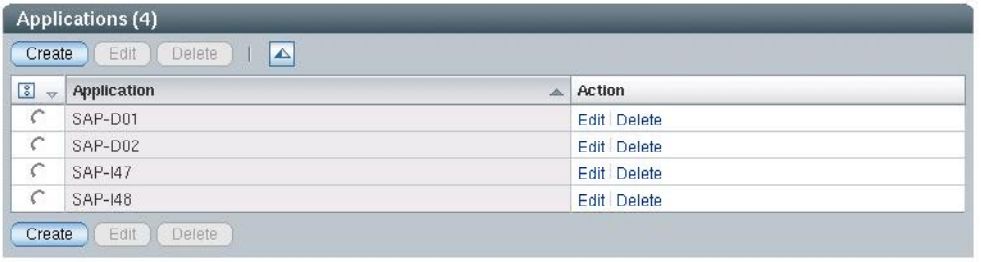

FIGURE 3–3 Applications

Maintain the applications of your landscape. An application is a group of components measured by the N1 AA Analyzer, see "Components" on page 24.

For example, the SAP system D01 is maintained here as the application `SAP-D01`. Later, you assign measured components to the application `SAP-D01`: An oracle database, an SAP central instance, and one or more SAP application servers.

The assignment of components to applications has an impact on how your MDX queries work. This impact is because each of them is based on the component/application model. For more information, see "Views" on page 26.

Note – Be careful when deleting an application. You lose all subsequent components and all related resource consumption data that have been collected by the Performance Collector on the N1 AA Clients and imported into the database. In order to keep these components, assign them to another application.

Components

Purpose	Maintain components of your N1 AA Analyzer landscape.
Menu	Administration ⇒ Analyzer ⇒ Customizing
Tasks for an initial basic setup	Mandatory - but can be done or changed at any time

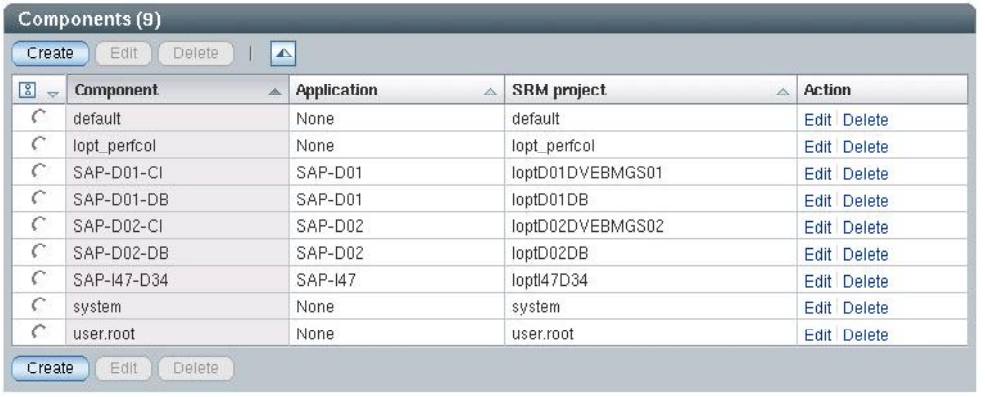

FIGURE 3–4 Components

Components are automatically recognized as running SRM projects in your landscape. Corresponding entries are created for each new component as performance data is imported. Normally, there is no need to create or delete an entry manually. It is important to maintain the following properties:

- Component: Name of the component, that is displayed in the GUI. The default name, as generated automatically by the data import job, is identical to the SRM project name. You are free to create meaningful names. For example, `SAP-D01-CI` for the central instance of the SAP system `D01`.
- Application: New components, as generated by the data import job, are assigned to the default application `None`. Assign the component to an existing application. For example, assign the component `SAP-D01-CI` to the application `SAP-D01`.
- SRM Project: This is the SRM project name that was measured. It is assumed, that the component is always started and running in this SRM Project. You do not have to modify this project, as long as this reflects reality.

You are free to assign and reassign components to applications in any way. However, the assignment does have an impact on how your MDX queries work. This is because each of them is based on the component/application model. For more information, see "Views" on page 26.

Best Practice

- Components that are unique in your landscape. For example, the central instance of SAP system `D01`.
  - Start them in a dedicated, landscape-wide unique, SRM-project. In this way, you measure the resource consumption of this component of the landscape separately because SRM-project = component.
  - Assign the component to an application other than `None`.

- Components according to the default SRM projects: system, default, and user.root. You can always find these components because the default SRM projects are always running.
  - Assign them to the application `None`.
- Additional server-bound system components. For example, a BMC Patrol Agent.
  - Default behavior: The load is included in one of the default SRM projects. Therefore, no additional component is be available.
  - If you would like to measure the load separately, start those components in a dedicated project but let the new component be assigned to the application `None`.

Caution – Deleting a component erases all related resource consumption data that has been collected by the Performance Collector on the N1 AA Clients and imported into the database.

Views

Overview

Once performance data (CPU and memory consumption of every component on every host of the N1 AA Landscape) are imported into the database, you can use MDX queries in the N1 AA Analyzer for reporting and graphical illustration.

You can use the predefined MDX queries delivered along with N1 AA, but you can also write your own queries.

Purpose	Reporting and graphical illustration of resource consumption. Capacity planing. Create, modify or extend reporting functionalities (MDX queries).
Menu	*Server group*⇒ Analyzer ⇒ Views
Use case	Daily business

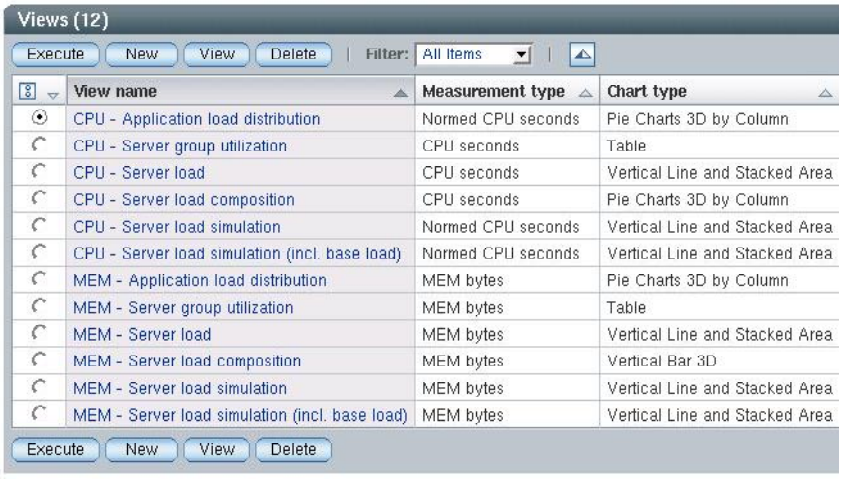

FIGURE 3–5 Views

MDX Queries

MDX is an abbreviation for Multi-Dimensional Expressions. This is a joint specification of the XML for Analysis founding members. MDX was created to query OLAP databases. MDX has become widely adopted within the realm of analytical applications.

To learn more about MDX, visit the Mondrian OLAP server home page at `http://mondrian.sf.net`. They have a comprehensive MDX specification including examples on how to write MDX queries.

Predefined MDX Queries

The N1 AA Analyzer comes with some predefined MDX queries allowing you to perform the most common analyses on your system landscape.

Note – These queries are read-only, you can neither modify nor delete them. These queries are available in every server group.

TABLE 3–3 MDX Queries

View name	Description	Benefits
CPU - Server group utilization	Get a CPU utilization overview of your server group. Key utilization values are displayed and the load is qualified. The qualification itself is based on threshold values that are customizable in the N1 AA. See "Basic Setup" on page 20. **Note –** Find details of this query and especially the meaning of the customizable parameters in the Appendix A. **Input/Usage** ■ Specify the time frame ■ Select none, one or more servers	Quickly identify CPU bottlenecks or oversupply.
MEM - Server group utilization	Get a memory utilization overview of your server group. Key utilization values are displayed and the load is qualified. The qualification itself is based on threshold values, that are customizable. See "Basic Setup" on page 20. **Note –** Find details of this query and especially the meaning of the customizable parameters in the Appendix A. **Input/Usage** ■ Specify the time frame ■ Select none, one, or more servers	Quickly identify memory bottlenecks or oversupply.

TABLE 3–3 MDX Queries *(Continued)*

View name	Description	Benefits
CPU - Application load distribution	CPU consumptions of all components belonging to the specified application are displayed in relation to each other. Data basis are the normalized CPU seconds in the specified time frame. **Input/Usage** ■ Specify the time frame ■ Select one application	Find the main CPU consuming components of an application.
MEM - Application load distribution	Memory consumptions of all components belonging to the specified application are displayed in relation to each other. **Input/Usage** ■ Specify the timeframe ■ Select one application	Find the main memory consuming components of an application.
CPU - Server load	Display the used CPU seconds of every component along the specified timeline. **Input/Usage** ■ Specify the time frame ■ Select one server	Find the CPU capacity situation on the server.
MEM - Server load	Display the used memory of every component along the specified timeline. **Input/Usage** ■ Specify the timeframe ■ Select one server	Find the memory capacity situation on the server.
CPU - Server load composition	CPU consumptions of all components running on the specified host are displayed in relation to each other. Data basis are the CPU seconds in the specified time frame. **Input/Usage** ■ Specify the timeframe ■ Select one server	Find the main CPU consuming components running on the server.

TABLE 3–3 MDX Queries *(Continued)*

View name	Description	Benefits
MEM - Server load composition	Average memory consumption, in bytes, of all components running on the specified host are displayed. The `None.Free` value represents the free physical memory capacity and can be negative. **Input/Usage** ■ Specify the time frame ■ Select one server	Find the main memory consuming components running on the server.
CPU - Server load simulation	What-If Analysis: Simulates the consolidation of application components on a target server. The output graphic displays the resulting CPU utilization on that server. The available server capacity is displayed as a horizontal line. All are based on the normalized CPU consumptions. **Input/Usage** ■ Specify the time frame ■ Select one target server ■ Select one or more application components	Reduce Risk: Simulate capacity situation before application consolidation.
MEM - Server load simulation	What-If Analysis: Simulates the consolidation of application components on a target server. The output graphic displays the resulting memory utilization on that server. The available server capacity is displayed as a horizontal line. All are based on the measured memory consumptions. **Input/Usage** ■ Specify the time frame ■ Select one target server ■ Select one or more application components	Reduce Risk: Simulate capacity situation before application consolidation.

TABLE 3–3 MDX Queries *(Continued)*

View name	Description	Benefits
CPU - Server load simulation (incl. base load)	Same as CPU – Server load simulation. The difference being that all component loads of application `None`except the `None.Free` value are left on the selected target server as some kind of base CPU consumption. **Input/Usage** ■ Specify the time frame ■ Select one target server ■ Select one or more application components	Reduce Risk: Simulate capacity situation before application consolidation.
MEM - Server load simulation (incl. base load)	Same as MEM – Server load simulation. The difference being that all component loads of application `None` except the `None.Free` value are left on the selected target host as some kind of base memory consumption. **Input/Usage** ■ Specify the time frame ■ Select one target server ■ Select one or more application components	Reduce Risk: Simulate capacity situation before application consolidation.

Create Your Own MDX Queries

You can create and modify your own MDX queries. See Appendix A for detailed information.

Healthcheck

Purpose Check N1 AA Analyzers consistency

Menu *servergroup* ⇒ Analyzer ⇒ Healthcheck

Use case Daily business

The healthcheck monitor provides a status overview of your landscape. The healthcheck monitor displays the status of the Performance Collector daemons that are located on the N1 AA clients. The healthcheck monitor also shows if the data that has been recorded and imported into the N1 AA database, contains gaps or not.

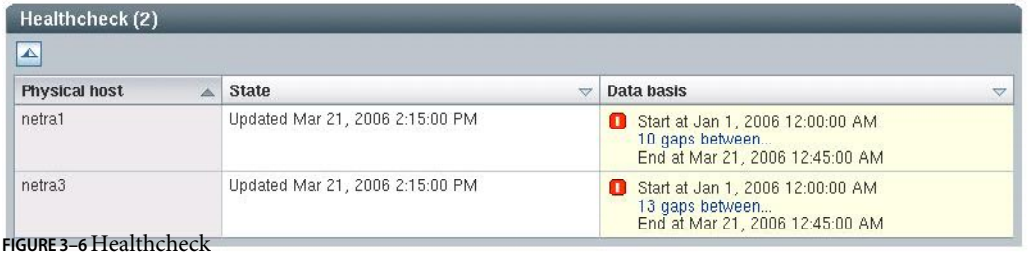

FIGURE 3–6 Healthcheck

Based on the health of these checklist items, the following state and data basis alerts can be displayed.

TABLE 3–4 States and Alerts

State	Description
Updated *timestamp*	All okay – The Performance Collector daemon is running and data is written at *timestamp*.
Updated *timestamp*	Minor – The Performance Collector daemon is running and data is written locally at *timestamp*, but no data is imported into the database. This could be a temporary problem that occurs prior to your first data import. However, check whether data imports are scheduled correctly. For more information, see "Basic Setup" on page 20.
Updated *timestamp*	Major: The Performance Collector daemon is running and data is written locally for the last time at *timestamp*, but the data in the database is older than one day. Check whether data imports are scheduled correctly. For more information, see "Basic Setup" on page 20.
Not updated since *timestamp*	Critical: Host is up, but the Performance Collector daemon has not written any local data since *timestamp*. There is a problem with the daemon. For more information, see the *Sun N1 Advanced Architecture for SAP Solutions 5.2.1 Installation Guide*.
Unknown	General: Host is down or Performance Collector data cannot be accessed. Since connection attempts are unsuccessful, no further information about the daemon state can be given. There could be a problem with the host, the daemon, or the network. For more information, see the *Sun N1 Advanced Architecture for SAP Solutions 5.2.1 Installation Guide*.

TABLE 3–5 Data Basis

Data Basis	Description
Start at *timestamp* End at *timestamp*	All okay: Data basis, in the database, extends from *timestamp* to *timestamp*.
 Start at *timestamp* Gap from *timestamp* to *timestamp* End at *timestamp*	Critical: Data basis, in the database, extends from *timestamp* to *timestamp*, but contains gaps from *timestamp* to *timestamp*. You will discover these gaps when executing analyses and displaying results. For more information, see "Views" on page 26.

Maintenance

Cube

Purpose Invalidate N1 AA Analyzer cube data cache

Menu Administration ⇒ Analyzer ⇒ Maintenance

Use case Exceptional circumstances

You can use the Invalidate button to clear the N1 AA Analyzer cube data cache in memory. This cache is built up during execution of N1 AA Analyzer analyses and significantly increases analysis performance. It is valid only while no change happens to the underlying N1 AA Analyzer data and repository data in the database. Generally, the N1 AA Analyzer does all the cache management itself, so there is no need to invalidate it manually. Only use this function in case you had changed the N1 AA database manually and want to get rid of the memory-mirrored old cache data.

Reorganizations

Purpose Manually archive or unarchive historical N1 AA Analyzer data

Menu Administration ⇒ Analyzer ⇒ Maintenance

Use case Weekly or monthly business. Only needed if no reorganization is already scheduled as job in the background.

By entering a start and end timestamp and pressing the Archive or Unarchive button, a reorganization is executed through a newly created `ArchiveUnarchiveJob` job. In general, keeping the N1 AA Analyzer database small, offers maximum analysis performance. However, besides all regularly scheduled reorganization runs you can move data into an archive table (technically between database tables `ResourceConsumption` and `ResourceConsumptionArchive`) without limitation. Additionally, to keep the database clean, you can export or import the archive and

vacuum the database using PostgreSQL database tools. Note that while an `ArchiveUnarchiveJob` is running, both the Archive and Unarchive buttons are disabled.

Imports

Purpose	Manually import N1 AA Analyzer data
Menu	Administration ⇒ Analyzer ⇒ Maintenance
Use case	Daily business. Only needed if no import is already scheduled as job in the background.

By clicking the Execute button, an N1 AA Analyzer data import is started. The data, that had been previously written locally by the performance collectors, is imported into the N1 AA database. If a regular data import is not scheduled by a persistent `AnalyseDataImportJob` job, a new transient job is created and executed automatically. For more information, see "Scheduler" on page 16. Otherwise, the persistent job is used for this execution. The advantage of this behavior is that all log lines of this manual data import job are appended to the jobs normal log along with the scheduled job lines and saved to the database. Note that while an `AnalyseDataImportJob` job is running, the Execute button appears disabled.

◆ ◆ ◆ CHAPTER 4

N1 AA Builder

This chapter describes the setup and usage of the N1 AA Builder module.

Overview

The Builder is based on the technology of the Sun N1 Service Provisioning System (N1 SPS). In that way, the Builder provides a well defined access to:

- OS Provisioning – Complete OS installation from a bare metal state.
- SW Provisioning – Installation of additional software on a running OS. For example, additional patches, packages, and files.

Customizing

Note – Note: Some necessary parameterisation for the N1 AA Builder is described in the general N1 AA setup. Those parameters are `none.general.sps.*`. For more information, see "General Customizing" on page 11.

This chapter describes additional customizing necessary for the N1 AA Builder.

Purpose	Customizing for N1 AA Builder
Menu	Administration ⇒ Builder ⇒ Customizing
Tasks for an initial basic setup	All changes are optional

FIGURE 4–1 N1 AA Builder Customizing

N1 AA Builder customization is done through these parameters. The following parameters are supported:

TABLE 4–1 Builder Customization Parameters

Parameter	Description	Example value
`none.builder.OSProvisioning.plan_timeout`	Maximum run time for a OS provisioning plan (in minutes).	60
`none.builder.OSProvisioning.plan_nativetimeout`	Maximum run time for native OS calls during execution of a OS provisioning plan (in minutes).	60
`none.builder.OSProvisioning.os_profile_types`	OS Profiles types supported by the N1 AA Builder. They must match or be a subset of OS Profiles types in SPS.	com.sun.n1osp#SolarisProfile, com.sun.n1osp#LinuxProfile, com.sun.n1osp#WindowsProfile
`none.builder.OSProvisioning.os_plugin_base_path`	Specifies the base folder of the OS Plug–in in SPS.	`/com/sun/n1osp`
`none.builder.SWProvisioning.n1aa_plan_path`	Base folder of plans in SPS available for N1 AA SW Provisioning.	`/com/sun/n1aa/swprovisioning`
`none.builder.SWProvisioning.plan_timeout`	Maximum run time for a SW provisioning plan (in minutes). **Note –** If you select more than one plan to be installed, this time out is valid for each one.	60

TABLE 4–1 Builder Customization Parameters *(Continued)*

Parameter	Description	Example value
`none.builder.SWProvisioning.plan_nativetimeout`	Maximum run time for native OS calls during execution of a SW provisioning plan (in minutes). **Note –** Note: If you select more than one plan to be installed, this time out is valid for each one.	60

OS Provisioning

Preparatory work

The contents for OS provisioning, OS Profiles and corresponding variable settings, have to be defined within the N1 SPS system.

1. Create all OS Profiles in the OSP Plug-in of the N1 SPS system, that you want to utilize within the N1 AA Builder.
2. **Optional:** Create one or more variable setting for every OS Profile if needed. Otherwise, only the default variable setting is be available.
3. Create all target hosts for OS Provisioning in the OSP Plug-in.

For detailed information on how to create OS Profiles or variable settings, see the *Sun N1 Service Provisioning System User's Guide for OS Provisioning Plug-In 3.1*.

Execute Provisioning

Purpose OS installation on specified target hosts

Menu Server Group ⇒ Builder ⇒ OS Provisioning

Use case Daily business

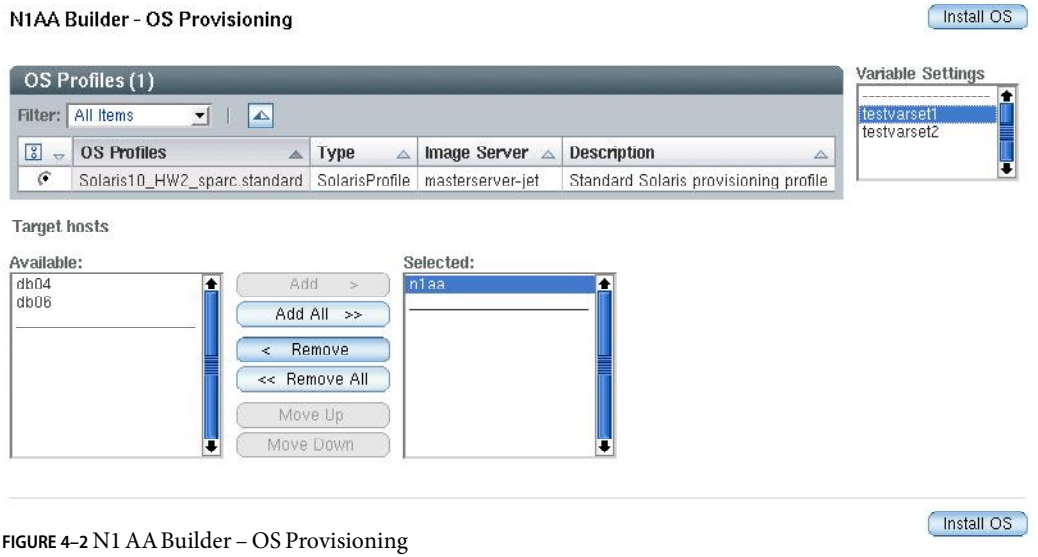

FIGURE 4–2 N1 AA Builder – OS Provisioning

1. Select one OS Profile in the OS Profiles table.
2. When you have selected the profile, the Variable Settings list displays all variable settings for that profile. The list might be empty.
3. Select one variable setting from the list or click ———— to select the default setting.
4. Select one or more target hosts.
5. Click the Install OS button to start the installation.

A job is scheduled in the background for the installation. One job is scheduled for each target host. See the job log for the progress of the installation. For more information on checking jobs and job logs, see "Scheduler" on page 16.

Software Provisioning

Preparatory Work

The contents for software provisioning, SPS plans, have to be defined in the N1 SPS system.

1. Create plans in the N1 SPS system.
2. Save the plans in the `/com/sun/n1aa/swprovisioning` directory. The N1 AA Builder only displays and executes plans that are stored in `/com/sun/n1aa/swprovisioning`.

Note – The N1 AA Builder does not support plan variables or component installation with specified variable settings. That means that every component used within a plan, is installed with its default values. You have to ensure that all variables of every component used within a plan contain suitable default values.

- Plans must not contain plan variables
- Every component within a plan is installed with its default values

Note – You have to ensure that all variables, of every component used within a plan, contain suitable default values.

For more information about creating plans and components, see the *Sun N1 Service Provisioning System 5.2 Plan and Component Developer's Guide*.

Execute Provisioning

Purpose Software installation on specified running target hosts

Menu Server Group ⇒ Builder ⇒ SW Provisioning

Use case Daily business

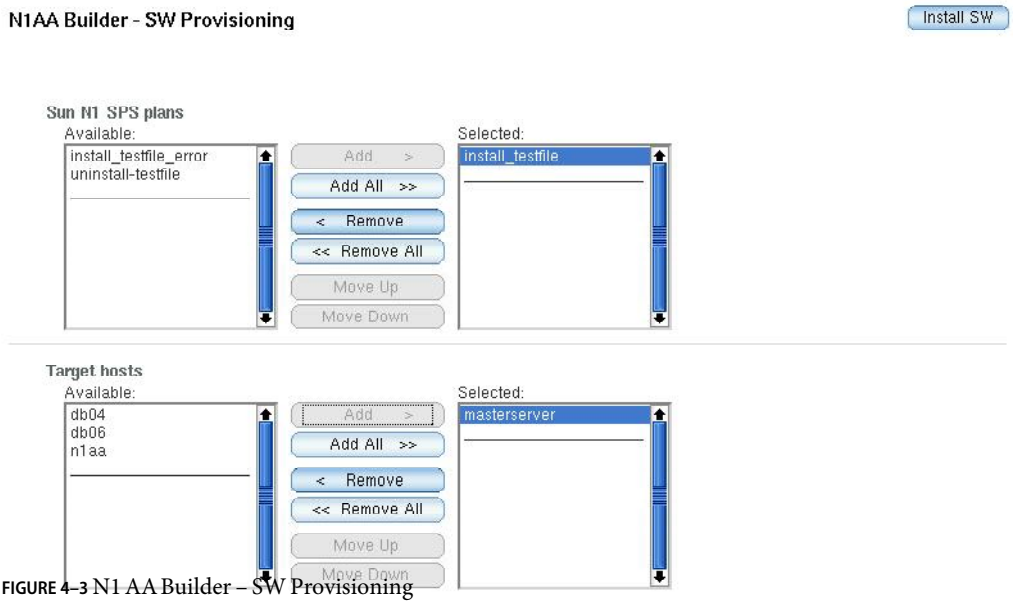

FIGURE 4–3 N1 AA Builder – SW Provisioning

1. Select one or more SPS plans to be installed. If you have selected more than one plan, you can specify the installation sequence by using the Move Up and Move Down buttons.

By selecting multiple plans, you are ensuring that they are executed sequentially. If a plan fails, the subsequent plans will not be run. If you want to run a plan independent of other plans, select only one plan and run it.

2. Specify one or more target hosts.
3. Click the Install SW button to start the installation.

A job is scheduled, in the background, for the installation; one job for every target host. View the job log for the progress of the installation. How to check jobs and job logs is described in "Scheduler" on page 16.

◆ ◆ ◆ CHAPTER 5

N1 AA Deployer

This chapter describes the setup and usage of the N1 AA Deployer module.

Overview

The N1 AA Deployer is designed to manage the services in your system landscape. The main functionalities are:

- Start and Stop resources. For example, an SAP Central Instance or complete resource groups
- Relocate resource groups from one server to another server
- Deploy or delete SAP Application Server instances

The N1 AA Deployer is based on the technology of the Sun N1 Service Provisioning System (N1 SPS). The system landscape you manage has to be defined completely within the N1 SPS SAP Plug-in. For more information, see the *Sun N1 Service Provisioning System User's Guide for SAP Plug-In 1.0* on how to setup resources and resource groups.

Customizing

This section describes additional customizing necessary for the N1 AA Deployer.

Note – The necessary SPS parameterisation should have already been done during the general N1 AA setup. The parameters `none.general.sps.*` have been specified. For more information, see "General Customizing" on page 11.

SAP RFC Parameter

Purpose	Settings for SAP RFC communication.
Menu	Administration ⇒ Deployer ⇒ Customizing
Tasks for an initial basic setup	Mandatory

While creating or removing SAP Application Instances, the existing Logon and RFC Servers Groups of the SAP System might be modified. The action of creating adds the instance to the SAP Systems Logon or RFC Server Groups. The action of removing deletes the instance from the SAP Systems Logon or RFC Server Groups.

N1 AA uses the RFC parameters to communicate with the SAP system. This is true for all SAP systems in your N1 AA landscape. You have to provide the same user ID and password in every SAP system.

TABLE 5–1 SAP RFC Parameters

Parameter	Description	Example value
`none.deployment.saprfc.client`	Logon client for RFC communication.	000
`none.deployment.saprfc.password`	Logon password for RFC communication. Password is stored encrypted in the N1 AA database.	n1aapwd
`none.deployment.saprfc.timeout`	Timeout value, in minutes, for RFC communication	2
`none.deployment.saprfc.user`	User for RFC communication. SAP user type `communications (CPI-C)` is sufficient.	n1aauser

Note – Regarding the previous parameters. You can add additional optional parameters by replacing `none` with *servergroup*. For example, *servergroup*`.deployment.saprfc.user`, where *servergroup* is the name of an existing server group. This overwrites the parameter value within the specified server group.

Timeout Parameter

Purpose	Settings the timeout values for Deployer actions
Menu	Administration ⇒ Deployer ⇒ Customizing
Tasks for an initial basic setup	Optional

This is where timeout values for Deployer actions are customized. The naming convention for these parameters is: `none.deployment.generaltimeout.`*resourcetype.action*. With:

- action = start: Timeout for starting a resource of type *resourcetype*
- action = stop: Timeout for stopping a resource of type *resourcetype*
- action = isonline: Timeout for checking if a resource of type *resourcetype* is online
- action = isoffline: Timeout for checking if a resource of type *resourcetype* is offline

---

Note – You can customize timeouts for a dedicated resource by adding parameters while replacing *resourcetype* with the dedicated *resourcename*. For example, *none.deployment.generaltimeout.P01CI.start* defines the start timeout for the resource P01CI. It overwrites the generic type–specific start timeout for SAP Central Instances `none.deployment.generaltimeout.SAPCentralInstance.start`.

TABLE 5–2 Timeout Parameters

Parameter	Description	Example Value
`none.deployment.generaltimeout.GDS.isoffline`	Timeout values, in minutes, for resources of type `GDS`.	1
`none.deployment.generaltimeout.GDS.isonline`		1
`none.deployment.generaltimeout.GDS.start`		5
`none.deployment.generaltimeout.GDS.stop`		5
`none.deployment.generaltimeout.HAStoragePlus.isoffline`	Timeout values, in minutes, for resources of type `HAStoragePlus`.	1
`none.deployment.generaltimeout.HAStoragePlus.isonline`		1
`none.deployment.generaltimeout.HAStoragePlus.start`		10
`none.deployment.generaltimeout.HAStoragePlus.stop`		10
`none.deployment.generaltimeout.LogicalHost.isoffline`	Timeout values, in minutes, for resources of type `LogicalHost`.	1
`none.deployment.generaltimeout.LogicalHost.isonline`		1
`none.deployment.generaltimeout.LogicalHost.start`		5
`none.deployment.generaltimeout.LogicalHost.stop`		5

TABLE 5–2 Timeout Parameters *(Continued)*

Parameter	Description	Example Value
none.deployment.generaltimeout.NfsMount.isoffline	Timeout values, in minutes, for resources of type NFSMount.	1
none.deployment.generaltimeout.NfsMount.isonline		1
none.deployment.generaltimeout.NfsMount.start		5
none.deployment.generaltimeout.NfsMount.stop		5
none.deployment.generaltimeout.NfsShare.isoffline	Timeout values, in minutes, for resources of type NFSShare.	1
none.deployment.generaltimeout.NfsShare.isonline		1
none.deployment.generaltimeout.NfsShare.start		5
none.deployment.generaltimeout.NfsShare.stop		5
none.deployment.generaltimeout.Oracle.isoffline	Timeout values, in minutes, for resources of type "Oracle".	1
none.deployment.generaltimeout.Oracle.isonline		1
none.deployment.generaltimeout.Oracle.start		10
none.deployment.generaltimeout.Oracle.stop		10
none.deployment.generaltimeout.SapAppServer.Install	Timeout values, in minutes, for resources of type AppServer.	10
none.deployment.generaltimeout.SapAppServer.isoffline		1
none.deployment.generaltimeout.SapAppServer.isonline		1
none.deployment.generaltimeout.SapAppServer.start		10
none.deployment.generaltimeout.SapAppServer.stop		10
none.deployment.generaltimeout.SapCentralInstance.isoffline	Timeout values, in minutes, for resources of type CentralInstance.	1
none.deployment.generaltimeout.SapCentralInstance.isonline		1
none.deployment.generaltimeout.SapCentralInstance.start		10
none.deployment.generaltimeout.SapCentralInstance.stop		10
none.deployment.generaltimeout.SapEnqueueServer.isoffline	Timeout values, in minutes, for resources of type EnqueueServer	1
none.deployment.generaltimeout.SapEnqueueServer.isonline		1
none.deployment.generaltimeout.SapEnqueueServer.start		5

TABLE 5–2 Timeout Parameters *(Continued)*

Parameter	Description	Example Value
`none.deployment.generaltimeout.SapEnqueueServer.stop`		5
`none.deployment.generaltimeout.SapMessageServer.isoffline`	Timeout values, in minutes, for resources of type `MessageServer`.	1
`none.deployment.generaltimeout.SapMessageServer.isonline`		1
`none.deployment.generaltimeout.SapMessageServer.start`		5
`none.deployment.generaltimeout.SapMessageServer.stop`		5

Note – Change all *.isonline and *.isoffline parameter values from 1 to 5. This is a absolutely necessary. Make these changes immediately after installation and before you use N1 AA.

If you see messages like `... Got timeoutout from db ...`, you might have to adjust the related timeout parameters.

Groups and Resources Management

This chapter explains all functions provided by the N1 AA Deployer to manage resources and resource groups.

Purpose Management of resources and resource groups. For example, start, stop, and relocate.

Menu *servergroup* ⇒ Deployer ⇒ Resource Groups

Use case Daily business

Go to *servergroup* ⇒ Deployment ⇒ Resource Groups. You arrive at the first main screen that lists all the available resource groups. This is the access point for all management functions.

FIGURE 5–1 N1 AA Deployer – Resource Groups

Select one dedicated group and click the View button . This brings you to the second main screen that lists all the available resources belonging to the selected group. This is the access point for all management functions. For example, start and stop the resources.

The following table lists all available functions on resource groups and resources in the N1 AA Deployer.

Caution – Be careful with the Force option of the Move function. You have to ensure that the resources are really offline. Otherwise, there is the danger that resources will be started twice. On the current host and on the target host.

TABLE 5–3 Functions

Function	Description	Mode	Input	Output
Resource Groups				
View Resource Groups	The initial screen of the Deployer. Shows all Resource Groups defined in N1 SPS.	Online in the foreground (GUI)	None	**Resource Name** Name of the resource group as defined in N1 SPS. **Status** Status of the Group on the host. The group status is derived from the status of its resources. ONLINE = All resources are online (up and running). OFFLINE = All resources are offline (down). PARTIAL = All resources are partially online. This means that all necessary processes are neither all online nor are they all offline. UNKNOWN = No status information for any resource available. UNREACHABLE = Host is unreachable. To be more technically correct, the SPS Remote Agent on this host is not reachable. MIXED = No uniform status of the resources (any mixed mode of statuses.

TABLE 5–4 Functions (continued)

Function	Description	Mode	Input	Output
Resource Groups				
View Resource Groups	The initial screen of the Deployer. Shows all Resource Groups defined in N1 SPS.	Online in the foreground (GUI)	None	**Last Update** Point in time when the status was last updated. N1 AA is not doing any active monitoring. Instead, the status is only updated when requested. That is: ■ Automatically after start, stop or relocate of resources/groups. ■ If refresh-functions have been performed by the user. **Host** Hostname on that the group is currently mastered as defined in N1 SPS. **Environment** Type of group as defined in N1 SPS. SPS = Managed directly by SPS SC = Group comes from Sun Cluster ACC = Group comes from SAP Adaptive Computing controller
Start	Start the resource group on the host. N1 AA calls the start plan of the group in N1 SPS. This starts all the resources of the group in the predefined sequence. For more details on the start mechanism, see the *Sun N1 Service Provisioning System User's Guide for SAP Plug-In 1.0*.	Job in the background. The GUI reflects the changes as soon as the job is finished and the GUI is refreshed by any user interaction.	**Resource Group** As selected in the GUI	The joblog shows details of the start: Success = Start of group was successfully executed within SPS. All resources are online. The group status, status of all resources, is set to online. Failed = Start of group was not successfully executed within SPS. At least one resource is not online now. The group status, status of all resources, is set to Unknown. Use the refresh functions to re–evalute the current status of the group.

TABLE 5–4 Functions (continued) *(Continued)*

Function	Description	Mode	Input	Output
Stop	Stop the resource group on the host. N1 AA calls the stop plan of the group in N1 SPS. This stops all resources of the group in the predefined sequence. For more details on the stop mechanism, see *Sun N1 Service Provisioning System User's Guide for SAP Plug-In 1.0*.	Job in the background. The GUI reflects the changes, as soon as the job is finished and the GUI is refreshed by any user interaction.	**Resource Group** As selected in the GUI	The joblog shows details of the stop: Success = Stop of group was successfully executed within SPS. All resources are offline now. The group status, status of all resources, is set to Offline. Failed = Stop of group was not successfully executed within SPS. At least one resource is not offline now. The group status, status of all resources, is set to Unknown. Use the refresh functions to reevalute the current status of the group.

TABLE 5–4 Functions (continued) *(Continued)*

Function	Description	Mode	Input	Output
Move	Relocate the resource group to a specified target host. This stops the group on the current host and starts the group on the specified target host.	Job in the background. The GUI reflects the changes as soon as the job is finished and the GUI is refreshed by any user interaction.	**Resource Group** As selected in the GUI **Target host** Physical hostname **Force Option** Ignores the fact that the group could not be stopped successfully on the current host and continues to start the group on the target host. This might be needed in special situations. For example, if the current host is down. **Caution –** Be careful with the Force option. You have to ensure that the resources are really offline. Otherwise, there is the danger that resources will be started twice. On the current host and on the target host.	The joblog shows details of the move: Success = Move of the group was successfully executed within SPS. All resources are online on the target host now. The group status, status of all resources, is set to Online. Failed = Move of the group was not successfully executed within SPS. The group status, status of all resources, is set to Unknown. Use the refresh functions to reevalute the current status of the group.
Refresh	Evaluate the current status of the resource group (every resource) on the host. N1 AA calls the appropriate `isonline` and `isoffline` plans in SPS.	Job in the background. The GUI reflects the changes as soon as the job is finished and the GUI is refreshed by any user interaction.	**Resource Group** As selected in the GUI	Updates the Status and Last Update information as displayed in the GUI. See the functions View Resource Groups and View Resources.
Resources				

TABLE 5–4 Functions (continued) *(Continued)*

Function	Description	Mode	Input	Output
View Resources	The screen shows all resources belonging to the specified group as defined in N1 SPS.	Online in the foreground (GUI)	**Resource Group** As selected in the GUI	**Resource Name** Name of the resource, as defined in N1 SPS. **Type** Type of the resource, as defined in N1 SPS. There are predefined resource types. For example, IP Address or SAP Central Instance, available with the SAP plug-in. **Status** Status of the resource on the host. ONLINE = Resource is online (up and running). OFFLINE = Resource is offline (down). PARTIAL = All resources are partially online. This means that all necessary processes are neither all online nor are they all offline. UNKNOWN = No status information for the resource is available. UNREACHABLE = Host is unreachable. To be more technically correct, the SPS Remote Agent on this host is not reachable.

TABLE 5–4 Functions (continued) *(Continued)*

Function	Description	Mode	Input	Output
				Last update Point in time, the status was last updated. N1 AA is not doing an active monitoring. Instead of that, the status will only be updated when requested. That is: ■ Automatically after start, stop, or relocate of resources or groups. ■ If refresh functions have been performed by the user. **Host** Hostname, on that the resource is currently mastered as defined in N1 SPS. **Environment**Type of the group, as defined in N1 SPS. SPS = Managed directly by SPS SC = Group comes from Sun Cluster ACC = Group comes from SAP Adaptive Computing controller
Start	Start the resource on the host. N1 AA calls the start plan of the resource in N1 SPS. For more details on the start mechanism, see *Sun N1 Service Provisioning System User's Guide for SAP Plug-In 1.0*.	Job in the background. The GUI reflects the changes as soon as the job is finished and the GUI is refreshed by any user interaction.	**Resource** As selected in the GUI	The joblog shows details of the start: Success = Start of resource was successfully executed within SPS. The resource status is set to Online. Failed = Start of resource was not successfully executed within SPS. The resource status is set to Unknown. Use the refresh functions to reevalute the current status of the resource.

TABLE 5–4 Functions (continued) *(Continued)*

Function	Description	Mode	Input	Output
Stop	Stop the resource on the host. N1 AA calls the stop plan of the resource in N1 SPS. For more details on the stop mechanism, see *Sun N1 Service Provisioning System User's Guide for SAP Plug-In 1.0*.	Job in the background. The GUI reflects the changes, as soon as the job is finished and the GUI is refreshed by any user interaction.	**Resource Group** As selected in the GUI	The job log shows details of the stop: Success = Stop of group was successfully executed within SPS. The resource status is set to Offline. Failed = Stop of resource was not successfully executed within SPS. The resource status is set to Unknown. Use the refresh functions to reevalute the current status of the resource.
Refresh	Evaluate the current status of the resource on the host. N1 AA calls the appropriate `isonline` and `isoffline` plans in SPS.	Job in the background. The GUI reflects the changes, as soon as the job is finished and the GUI is refreshed by any user interaction.	**Resource Group** As selected in the GUI	Updates the `Status` and `Last Update` information as displayed in the GUI. See the functions View Resource Groups and View Resources.
View Details	Shows the technical details of the resource definition.	Online in the foreground (GUI)	**Resource** As selected in the GUI	Technical details of the resource as defined in N1 SPS.

SAP Application Server Provisioning

N1 AA Deployer supports the provisioning of SAP Application Servers for existing SAP Systems.

Precondition:

- The SAP System has to be existing and defined within the N1 SPS SAP Plug-in.
- If you want to use a *custom exception file* for the deployment process, the custom exception file must have been previously created within SPS.

For technical background information on SAP Application Sever provisioning, see "Creating, Managing and Installing SAP Application Servers" in *Sun N1 Service Provisioning System User's Guide for SAP Plug-In 1.0*.

Deploy New SAP Application Server

Purpose Install a new SAP Application Server for an existing SAP system

Menu *servergroup* ⇒ Deployer ⇒ Deploy new SAP Application Server

Use case Daily business

- Description

  This function installs a new SAP Application Server for an existing SAP system:

  - Creates new SAP Application Server including logical host
  - Adjusts the SAP RFC Server Groups and Logon Groups
  - Creates appropriate resource group and resources, SAP Application Server and logical host, within the SPS repository
  - Starts the new logical host, IP address, and SAP Application Server

- Mode

  Job in the background. The GUI reflects the changes as soon as the job is finished. The GUI is refreshed by any user interaction.

- Input

  Necessary input parameters describe the new SAP instance to install.

  - SAP System: Select one of the existing SAP systems. Resources of type `SAP Central Instance` in the server group.
  - Instance number: Instance number for the new application server.
  - Physical host: Install and start the new instance on this physical host. Select one of the existing hosts in the server group.
  - Logical host: IP name for the new application server. Specify any existing but unused IP name.
  - Size: Category of memory size, small, medium, or large, for the new instance profile.
  - Profile parameter file: Choose an exception file for the deployment process. The exception file defines dedicated parameters to be used for the new Instance Profile of the SAP Application Instance:

    - Choose either the default exception file which is represented by the entry `----------`.
    - Choose a custom exception file that was previously created within SPS

      For more information on exception files, see "Creating, Managing and Installing SAP Application Servers" in *Sun N1 Service Provisioning System User's Guide for SAP Plug-In 1.0*.

  - SRM Project: (Optional) If you specify an existing SRM project, N1 AA always starts the SAP application server in this SRM project. If you are using N1 AA Analyzer, this is a key requirement. If you do not specify an SRM project, the `default` project is used.

If N1 SPS is able to connect to the specified SAP system, you see a list of available SAP Logon Groups and RFC Server Groups on the system. Otherwise, you see a warning. You can decide to continue or not continue. Select any of these groups. N1 AA adds the new SAP instance to these groups, if the installation finishes successfully.

- Output

  The joblog shows details of the actions:

  - The installation and starting of the new components (SAP Application Server and logical host).
  - The adjustments of the SAP RFC Server Groups and Logon Groups.
  - The adjustments within the SPS repository.

Remove SAP Application Server

Purpose Remove an existing SAP Application Server

Menu *servergroup* ⇒ Deployer ⇒ Remove SAP Application Server

Use case Daily business

- Description

  This function removes a complete resource group (SAP Application Server and logical host), that has been previously installed by the `Deploy new SAP Application Server` function of the N1 AA Deployer.

  - Stops and deletes all SAP Application Servers and logical hosts in this group, that have been created by the function `Deploy new SAP Application Server` if not modified manually by the user. This equates to one SAP Application Server and one logical host.
  - Removes the deleted SAP Application Servers from all SAP RFC groups or Logon groups.
  - Deletes the corresponding resource group and resources within the SPS repository.

- Mode

  Job in the background. The GUI reflects the changes as soon as the job is finished. The GUI is refreshed by any user interaction.

- Input

  Necessary input parameters specify the resource group, that should be removed.

  - Select one of the existing server groups . The selection is limited to those groups, that have been previously created by the function Deploy New SAP Application Server of the N1 AA Deployer.

- Output

  The job log shows details of the actions:

  - Stopping and deleting the SAP Application Servers and logical hosts.
  - The adjustments of the SAP RFC Server Groups and Logon Groups.

- The adjustments within the SPS repository.

◆ ◆ ◆ CHAPTER 6

N1 AA Command Line Interface

The following tables describe the Command Line Interface (CLI) commands for N1 AA. The CLI provides access to most of the functions in N1 AA.

Return code All functions return 0 if successful, otherwise the functions return a 1.

Note – The path to `n1aa_exec` is `/opt/SUNWn1aa/cli/n1aa_exec`. Execute the commands as OS user `noaccess`.

Introduction

You can call every command with the optional parameter `loglevel`=*loglevel*, where *loglevel* can be DEBUG, INFO, SUCCESS, WARNING, ERROR, FATAL, or PANIC. This parameter overwrites the `loglevel` defined by the N1 AA parameter `none.general.n1aa.loglevel`. For more information, see "Customizing" on page 11.

TABLE 6–1 Summary of N1 AA Manager Commands

Function	CLI Syntax	Description
List server groups	`./n1aa_exec GetServerGroups`	Lists the defined server groups. **Input parameters** – **Output** Column1: Unique ID of Server Group Column2: Name of Server Group

TABLE 6–1 Summary of N1 AA Manager Commands *(Continued)*

Function	CLI Syntax	Description
List hosts	`./n1aa_exec GetServerGroupHosts servergroup=`*ID*	Lists the defined physical hosts of the specified sever group. **Input parameters** `servergroup`: ID of Server Group as returned from CLI command `GetServerGroups` **Output** Column1: Hostname

For more information, see "Customizing" on page 11.

TABLE 6–2 Summary of N1 AA Analyzer Commands

Function	CLI Syntax	Description
Data import	`n1aa_exec analyzer.AnalyseDataImportJob`	Imports the performance data. **Input parameters** – **Output** –

TABLE 6–2 Summary of N1 AA Analyzer Commands *(Continued)*

Function	CLI Syntax	Description
Data reorganization (archive data)	`n1aa_exec analyzer.ArchiveUnarchiveJob a` *start end*	Executes the reorganization of the performance data from cube to archive. *start* and *end* define the data interval. **Input parameters** *start*: timestamp literal *end*: timestamp literal **Output** – Examples: ■ This archives all data for the year 2004: `n1aa_exec analyzer.ArchiveUnarchiveJob a "timestamp '2004-01-01 00:00:00'" "timestamp '2004-12-31 23:59:59'"` ■ This archives all data from year 2004 and older: `n1aa_exec analyzer.ArchiveUnarchiveJob a "timestamp '-infinity'" "timestamp '2004-12-31 23:59:59'"` ■ This archives all data older than one year: `n1aa_exec analyzer.ArchiveUnarchiveJob a "timestamp '-infinity'" "timestamp without time zone 'now' - interval '12 month'"`
Data reorganization (unarchive data)	`n1aa_exec analyzer.ArchiveUnarchiveJob u` *start end*	Executes the reorganization of the performance data from archive back to cube. *start* and *end* define the data interval. **Input parameters** *start*: timestamp literal *end*: timestamp literal **Output** – Examples: See preceding examples.

For more information, see "Maintenance" on page 33.

TABLE 6–3 Summary of N1 AA Builder Commands

Function	CLI Syntax	Description
List OS Profiles	`n1aa_exec builder.ListOSProfiles`	Lists the available OS Profiles. **Input parameters** – **Output** Column1: Full pathname of OS Profile Column2: OS Profile type
List variable settings	`n1aa_exec builder.ListVariableSettings profile_name=`*profile_name*	Lists the available Variable Settings for the specified OS Profile component. **Input parameters** *profile_name:* Complete path of profile name as returned from CLI command `builder.ListOSProfiles`. **Output** Column1: Name of Variable Setting

TABLE 6–3 Summary of N1 AA Builder Commands *(Continued)*

Function	CLI Syntax	Description

Install OS	`n1aa_exec builder.ProvisionOS profile_name=`*OS Profile* `profile_type=`*type* `[varset_name=`*variable settings*`]` `targethost=`*target host* `[p_timeout=`*minutes*`]` `[n_timeout=`*minutes*`]`	Executes the OS installation with the specified OS Profile component using supplied Variable Settings on the specified target hosts.
		Input parameters
		`profile_name`: Name of OS Profile as delivered from CLI command `builder.ListOSProfiles`
		`profile_type`: Type of OS Profile as delivered from CLI command `builder.ListOSProfiles`
		`varset_name` (optional): Name of Variable Setting as delivered from CLI command `builder.ListVariableSettings`. If not specified, the OS Profile's default settings will be used.
		`targethost`: Target host for OS installation as delivered from CLI command `builder.GetOSProvisioningHosts`.
		`p_timeout` (optional): Installation timeout in minutes. If not specified, the pre-customized value `none.builder.OSProvisioning.plan_timeout` is taken.
		`n_timeout` (optional): Timeout for native OS calls in minutes. If not specified, the precustomized value `none.builder.OSProvisioning.plan_nativetimeou` is taken.
		Output
		–

List SPS Plans	`n1aa_exec builder.ListSWPlans`	Lists all SPS Plans available for N1 AA. **Input parameters** – **Output** Column1: Name of SPS plan

TABLE 6–3 Summary of N1 AA Builder Commands *(Continued)*

Function	CLI Syntax	Description
Install SW	`n1aa_exec builder.RunSWPlans plan_names=`*SPS Plan*`[,...] targethost=`*target host*` [p_timeout=`*minutes*`] [n_timeout=`*minutes*`]`	Executes the SW installation with the specified SPS plans on the specified target hosts. **Input parameters** `plan_names`: Name of SPS Plans as delivered from CLI command `builder.ListSWPlans`. Multiple plan names are separated by commas. `targethost`: Target host for SW installation as delivered from CLI command `builder.GetSWProvisioningHosts`. `p_timeout` (optional): Installation timeout in minutes. If not specified, the pre-customized value `none.builder.SWProvisioning.plan_timeout` is taken. `n_timeout` (optional): Timeout for native OS calls in minutes. If not specified, the pre–customized value `none.builder.SWProvisioning.plan_nativetimeout` is taken. **Output** -

Get OS provisioning hosts	`n1aa_exec builder.GetOSProvisioningHosts servergroup=`*ID*`| servergroupname=`*name*	List all hosts allowed for OS provisioning in the specified server group.
		Input parameters
		`servergroup`: ID of Server Group as returned from CLI command `GetServerGroups`
		`servergroupname`: Name of Server Group as returned from CLI command `GetServerGroups`
		Output
		Column1: Hostname

TABLE 6–3 Summary of N1 AA Builder Commands *(Continued)*

Function	CLI Syntax	Description

Get SW provisioning hosts	`n1aa_exec builder.GetSWProvisioningHosts servergroup=`*ID*`| servergroupname=`*name*	List all hosts allowed for SW provisioning in the specified server group.
		Input parameters
		`servergroup`: ID of Server Group as returned from CLI command `GetServerGroups`
		`servergroupname`: Name of Server Group as returned from CLI command `GetServerGroups`
		Output
		Column1: Hostname

For more information, see Chapter 4.

TABLE 6–4 Summary of N1 AA Deployer – Managing Resource Groups Commands

Function	CLI Syntax	Description
View Resource Groups	`n1aa_exec deployment.ListResourceGroups servergroup=`*ID*	List resource groups of the specified server group. **Input parameters** `servergroup`: ID of Server Group as returned from CLI command `GetServerGroups` **Output** Column1: Name of Resource Group Column2: Status (ONLINE, OFFLINE, ...) Column3: Last update timestamp Column4: Hostname where Resource Group is currently hosted Column5: Environment type Column6: Unique ID of Resource Group **Note –** This ID will change after moving the Resource Group to another target server.

TABLE 6–4 Summary of N1 AA Deployer – Managing Resource Groups Commands *(Continued)*

Function	CLI Syntax	Description
Start	`n1aa_exec deployment.ModifyResourceGroup resourcegroupid=`*ID* `command=start`	Start the resource group. **Input parameters** `resourcegroupid`: ID of Resource Group as returned from CLI command `deployment.ListResourceGroups` **Output** –
Stop	`n1aa_exec deployment.ModifyResourceGroup resourcegroupid=`*ID* `command=stop`	Stop the resource group. **Input parameters** `resourcegroupid`: ID of Resource Group as returned from CLI command `deployment.ListResourceGroups` **Output** –

Move	`n1aa_exec deployment.MoveResourceGroup resourcegroupid=`*ID* `targethost=`*hostname* `force=true|false`	Relocate the resource group to a specified target host.
		Input parameters
		`resourcegroupid`: ID of Resource Group as returned from CLI command `deployment.ListResourceGroups`
		`targethost`: Hostname as returned from CLI command `GetServerGroupHosts`
		`force`: If set to true, relocation will continue, even if the stop on the source host was not successful. Otherwise, if set to false, the operation stops.
		Output
		–

TABLE 6–4 Summary of N1 AA Deployer – Managing Resource Groups Commands *(Continued)*

Function	CLI Syntax	Description
Refresh	`n1aa_exec deployment.ModifyResourceGroup resourcegroupid=`*ID* `command=refresh`	Evaluate and update the current status of the resource group (every resource) on the host. **Input parameters** `resourcegroupid`: ID of Resource Group as returned from CLI command `deployment.ListResourceGroups`. **Output** –
Refresh server group	`n1aa_exec deployment.ModifyResourceGroup servergroup=`*ID* `command=refreshall`	Similar to the `Refresh()` function, but for all resource groups in the server group. **Input parameters** `servergroup`: ID of Resource Group as returned from CLI command `GetServerGroups`. **Output** –

For more information, see "Groups and Resources Management" on page 45.

TABLE 6–5 Summary of N1 AA Deployer – Managing Resources Commands

Function	CLI Syntax	Description
View Resources	`n1aa_exec deployment.ListResources resourcegroupid=`*ID*	List resources of the specified resource group. **Input parameters** `resourcegroupid`: ID of Resource Group as returned from CLI command `deployment.ListResourceGroups` **Output** Column1: Name of Resource Column2: Type of Resource Column3: Status Column4: Last update timestamp Column5: Hostname where resource is currently hosted Column6: Environment type Column7: Unique ID of resource **Note** – This ID will change after moving the Resource Group to another target server.
Start	`n1aa_exec deployment.ModifyResource resourceid=`*ID* `command=start`	Start the resource. **Input parameters** `resourceid`: ID of Resource as returned from CLI command `deployment.ListResources` **Output** -
Stop	`n1aa_exec deployment.ModifyResource resourceid=`*ID* `command=stop`	Stop the resource. **Input parameters** `resourceid`: ID of Resource as returned from CLI command `deployment.ListResources` **Output** -

TABLE 6–5 Summary of N1 AA Deployer – Managing Resources Commands *(Continued)*

Function	CLI Syntax	Description
Refresh	`n1aa_exec deployment.ModifyResource resourceid=`*ID* `command=refresh`	Evaluate the current status of the resource on the host. **Input parameters** `resourceid`: ID of Resource as returned from CLI command `deployment.ListResources` **Output** -
View Details	`n1aa_exec deployment.ListResourceDetails resourceid=`*ID*	Shows the technical details of the resource definition. **Input parameters** `resourceid`: ID of Resource as returned from CLI command `deployment.ListResources` **Output** Column1: Name of resource attribute Column2: Value of resource attribute

TABLE 6–6 Summary of N1 AA Deployer – SAP Application Server Provisioning Commands

Function	CLI Syntax	Description
List CI's	`n1aa_exec deployment.ListSAPCIs servergroup=`*ID*	Lists the defined SAP CI's, type CentralInstance, and message servers, type MessageServer, within the specified server group. **Input parameters** `servergroup`: ID of Server Group as returned from CLI command `GetServerGroups` **Output** Column1: Unique ID of Resource **Note –** This ID changes after moving the Resource Group to another target server. Column2: Name of Resource Column3: Name of Resource Group Column4: Status Column5: Last update timestamp
List custom exception files	`n1aa_exec deployment.ListCustomExceptionFiles`	Lists the defined custom exception files. **Input parameters** - **Output** Column1: Name of the custom exception file

TABLE 6–6 Summary of N1 AA Deployer – SAP Application Server Provisioning Commands *(Continued)*

Function	CLI Syntax	Description
Deploy	`n1aa_exec deployment.DeploySAPAppServer resourceid=`*ID* `logicalhost=`*hostname* `interface=`*name* `interfacetype=C|M targethost=`*hostname* `sysnum=`*instanceno* `profile=small|medium|large [exceptionfile=`*filename*`][srmproject=`*SRMProject*`]`	Install a new SAP Application Server for an existing SAP system. **Input parameters** `resourceid`: ID of the existing SAP target system as returned from CLI command `deployment.ListSAPCIs` `logicalhost`: IP Name to use for the new Application Server. It must be an existing and resolvable IP name in the network. `interface`: Name of the interface or ipmp-group to be used. `interfacetype`: If set to C, then parameter interface is interpreted as physical NIC. If set to M, then parameter interface is interpreted as ipmp-group. `targethost`: Hostname as returned from CLI command `GetServerGroupHosts`. `sysnum`: Instance number for the new sap instance. `profile`: Defines the memory category for the new instance profile (small, medium, or large.) `exceptionfile` (optional): Name of a custom exception file as returned from CLI command `ListCustomExceptionFiles`. If not specified, the default exception file is used. `srmproject` (optional): Solaris SRM project for the new instance. If specified, this project has exist on the OS level. If not specified, the SRM default is used. **Output** -

TABLE 6–6 Summary of N1 AA Deployer – SAP Application Server Provisioning Commands *(Continued)*

Function	CLI Syntax	Description
Deploy and modify Logon or RFC groups	`n1aa_exec deployment.DeploySAPAppServerWithServerGroup resourceid=`*ID* `logicalhost=`*hostname* `interface=`*name* `interfacetype=C|M targethost=`*hostname*`sysnum=`*instanceno* `profile=small|medium|large [srmproject=`*SRMProject*`] saplogongrouplist=`*group[,...]* `sapservergrouplist=`*group[,...]* `instance=`*instance*	Installs a new SAP Application Server on an existing SAP system and adds the new SAP instance to the specified Logon and RFC groups. **Input parameters** See `deployment.DeploySAPAppServer`, additionally: `saplogongrouplist`: SAP Logon Groups (comma separated list), to which this new instance should be added. As returned by CLI command `deployment.ListSAPServerGroups`. `sapservergrouplist`: SAP RFC Server Groups , comma separated list, to which this new instance should be added. As returned by CLI command `deployment.ListSAPServerGroups`. `instance`: String of the new SAP instance to be added to the Logon or RFC groups. Format must be exactly as defined in the SAP system, *log.hostname_SID_instance#*. Example: lhost0_I47_20. **Output** –
List deployed resource groups	`n1aa_exec deployment.ListResourceGroups servergroup=`*ID*`removeable=true`	Lists those resource groups that have been created within the N1 AA `Deploy new Application Server()` function. **Input parameters** `servergroup`: ID of Server Group as returned from CLI command `GetServerGroups` **Output** Same as previous CLI command `deployment.ListResourceGroups`. See row "View Resource Groups".

TABLE 6–6 Summary of N1 AA Deployer – SAP Application Server Provisioning Commands *(Continued)*

Function	CLI Syntax	Description
Remove	`n1aa_exec deployment.RemoveSAPAppServer resourceid=`*ID*	Remove an existing SAP Application Server (complete resource group including SAP Application Server instances and IP addresses). **Input parameters** `resourceid`: ID of the existing Resource Group as returned from CLI command `deployment.ListResourceGroups`, with `removeable`=true. **Output** -
List Logon/RFC groups	`n1aa_exec deployment.ListSAPServerGroups resourceid=`*ID*`grouptyp=S|L`	List the Logon or RFC groups of the specified SAP system. **Input parameters** `resourceid`: ID of the existing SAP target system as returned from CLI command `deployment.ListSAPCIs` `grouptyp`: L for Logon Groups, S for RFC Server Groups. **Output** Column1: Name of SAP Logon Group or RFC Server Group as defined in the SAP system.

TABLE 6–6 Summary of N1 AA Deployer – SAP Application Server Provisioning Commands *(Continued)*

Function	CLI Syntax	Description
List Instances	`n1aa_exec deployment.ListInstancesOfServerGroup resourceid=`*ID* `grouptyp=S|L sapservergroup=`*group*	List the SAP instances of the specified Logon/RFC group. **Input parameters** `resourceid`: ID of the existing SAP target system as returned from CLI command `deployment.ListSAPCIs` `grouptyp`: L for Logon Groups, S for RFC Server Groups. `sapservergroup`: Logon/RFC Server Group as returned by CLI command `deployment.ListSAPServerGroups`. **Output** Column1: Name of SAP instance belonging to the specified group as defined in the SAP system.
Modify Logon or RFC groups	`n1aa_exec deployment.ModifySAPServerGroups resourceid=`*ID* `grouptyp=L|S instance=`*instance* `command=R|I sapservergrouplist=`*group[,...]*	Add or remove an SAP instance to or from the specified Logon or RFC group. **Input parameters** `resourceid`: ID of the existing SAP target system as returned from CLI command `deployment.ListSAPCIs` `grouptyp`: L for Logon Groups, S for RFC Server Groups. `instance`: String of the SAP instance to be added or removed. Format as defined in the SAP system *log.hostname_SID_instance#*. For example, lhost0_I47_20 `command`: R for remove. I for insert. `sapservergrouplist`: Logon/RFC Server Groups (comma separated list), to which the instance should be added or removed as returned by CLI command `deployment.ListSAPServerGroups`. **Output** -

◆ ◆ ◆ APPENDIX A

MDX Queries

N1 AA Analyzer: Server Group Utilization

This section gives a detailed explanation of both MDX queries:

- CPU - Server group utilization
- MEM - Server group utilization

Based on the measured CPU/Memory utilization, the total host utilization within the specified timeframe is displayed and qualified. For example, low, good, or high host utilization. The qualification is based on customizable threshold values *%cpulow* and *%cpuhigh*.

	Measures									
ServerGroup	Avg util	Peak util	Avg %	Max %	Low #	Good #	High #	Used	Free	Total
netra1	low	some	11.5	37.4	88	8	0	9,923	76,477	86,400
netra3	low	some	1.4	4.9	96	0	0	1,202	85,198	86,400

FIGURE A–1 Example of CPU Utilization

Looking at the host CPU utilization of `netra1` in the specified time frame:

- Avg util: The average utilization was low, because the measured average of 11.5%, see column `Avg%`, is less than the threshold of 30% (the value of *%cpulow*).
- Peak util: There have been only *some* utilization peaks. That means less than 5% (the value of *%cpupeakslower*) of all measured values are greater than the threshold of 75% (the value of *%cpuhigh*).
- Avg % = The average utilization was 11.5%.
- Max % = The maximal utilization was 37.4%.
- Low # = 88 values were less than the threshold of 30% (the value of *%cpulow*).
- Good # = 8 values are in between 30% and 75% (the value of *%cpulow* and *%cpuhigh*).
- High # = 0 values are greater than the threshold of 75% (the value of *%cpuhigh*).

- Used = 9,923 CPU seconds have been utilized. (Absolute number of used CPU seconds).
- Free = 76,477 CPU seconds have been free. (Absolute number of unused CPU seconds).
- Total = 86,400 CPU seconds have been available. (Absolute number of available CPU seconds = Used + Free).

	Measures									
ServerGroup	Avg util	Peak util	Avg %	Max %	Low #	Good #	High #	Used #	Free #	Total #
netra1	high	many	294.3	301.9	0	0	96	987,382,016	-651,837,696	335,544,320
netra3	low	some	19.1	19.8	96	0	0	102,471,595	434,399,317	536,870,912

FIGURE A-2 Example of Memory Utilization

Looking at the host memory utilization of `netra1` in the specified timeframe:

- Avg util: The average utilization was high, because the measured average of 294.3% (see column Avg%) is greater than the threshold of 75% (the value of *%memhigh*).
- Peak util: There have been many utilization peaks. That means more than 10% (the value of *%mempeaksupper*) of all measured values are greater than the threshold of 75% (the value of *%memhigh*).
- Avg % = The average utilization was 294.3%.
- Max % = The maximal utilization was 301.9%.
- Low # = 0 values are less than the threshold of 30% (the value of *%memlow*).
- Good # = 0 values (of all measured values) are in between 30% and 75% (the value of *%memlow* and *%memhigh*).
- High # = 96 values (of all measured values) are greater than the threshold of 75% (the value of *%memhigh*).
- Used 0 = On average 987,382,016 bytes have been allocated.
- Free 0 = On average –651,837,696 bytes have been free. A negative value indicates that more memory was allocated from the applications than was available as RAM.
- Total 0 = On average 335,544,320 bytes have been available memory as RAM (the total of Used + Free).

The parameters mentioned previously are maintained in N1 AA Analyzer customizing. See "Basic Setup" on page 20. The following table summarizes all available parameters within both MDX queries.

Note – All the parameters given in the table below can be used for server group spanning or server group specific parameterisation.

- The parameter `none.analyzer.constants.XY` is mandatory and server group spanning valid.
- The parameter *servergroup*`.analyzer.constants.XY` is optional and server group specific valid. It overwrites the parameter `none.analyzer.constants.XY` within this specific server group.

TABLE A–1 Server Group Parameters

Constant	Description	Default Value
`none.analyzer.constants.%cpulow`	Lower threshold for CPU average utilization in percentage. If the measured average host utilization is smaller than *%cpulow*, the qualification is low.	30
`none.analyzer.constants.%memlow`	Lower threshold for memory average utilization in percentage. If the measured average host utilization is smaller than *%memlow*, the qualification is low.	30
`none.analyzer.constants.%cpuhigh`	Upper threshold for CPU average utilization in percentage. If the measured average host utilization is in between *%cpulow* and *%cpuhigh*, the qualification is good. If the measured average host utilization is greater than *%cpuhigh*, the qualification is high.	75
`none.analyzer.constants.%memhigh`	Upper threshold for memory average utilization in percentage. If the measured average host utilization is in between *%memlow* and *%memhigh*, the qualification is good. If the measured average host utilization is greater than *%memhigh*, the qualification is high.	75
`none.analyzer.constants.%cpupeakslower`	Lower threshold for CPU peak utilization in percentage. If less than *%cpupeakslower* percent of all measured host utilization values are greater than *%cpuhigh*, the qualification is some.	5
`none.analyzer.constants.%mempeakslower`	Lower threshold for memory peak utilization in percentage. If less than *%mempeakslower* percent of all measured host utilization values are greater than *%memhigh*, the qualification is some.	5

TABLE A–1 Server Group Parameters *(Continued)*

Constant	Description	Default Value
`none.analyzer.constants.%cpupeaksupper`	Upper threshold for CPU peak utilization in percentage. If more than *%cpupeakslower* AND less than *%cpupeaksupper* percent of all measured host utilization values are greater than *%cpuhigh*, the qualification is more. If more than *%cpupeaksupper* percent of all measured host utilization values are greater than *%cpuhigh*, the qualification is many.	10
`none.analyzer.constants.%mempeaksupper`	Upper threshold for memory peak utilization in percentage. If more than *%mempeakslower* and less than *%mempeaksupper* percent of all measured host utilization values are greater than *%memhigh*, the qualification is more. If more than *%mempeaksupper* percent of all measured host utilization values are greater than *%memhigh*, the qualification is many.	10

N1 AA Analyzer: Create Your Own MDX Queries

This section describes how to create your own MDX queries.

Note – Created MDX queries are available within the server group only.

These MDX queries are based on the underlying database model. The following picture gives an overview on this model. The fields in cursive letters will be available as measurands for your MDX Queries within the cube.

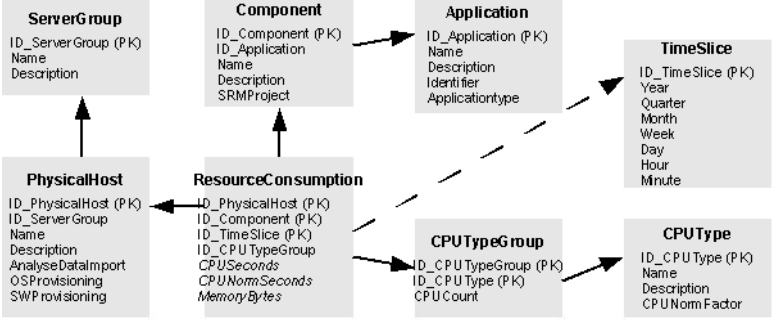

FIGURE A–3 Overview of MDX Queries

Every measurand taken by the Performance Collector on an N1 AA Client leads to a set of

self-contained entries in the Resource Consumption table. The measurand is taken every 15 minutes. Each of these entries describes the resource consumption of exactly one measured application component at the given point in time.

TABLE A–2 Resource Consumption

ID_PhysicalHost	ID_Component	ID_TimeSlice	ID_CPUTypeGroup	CPUSeconds	CPUNormSeconds	MemoryBytes
1	6	2006-01-01 00:30:00	1	0.83	1.66	105553920
1	4	2006-01-01 00:30:00	1	157.87	315.74	177684480
1	3	2006-01-01 00:30:00	1	48.86	97.72	703455232
1	22	2006-01-01 00:30:00	1	10.41	20.82	4481024
1	5	2006-01-01 00:30:00	1	9.30	18.60	0
1	1	2006-01-01 00:30:00	1	658.00	1316.00	–655630336
1	2	2006-01-01 00:30:00	1	14.73	29.46	0

(with ID_Component 1 = Free; 2 = Other; 3 = SAP-D01-CI; 4 = SAP-D01-DB; 5 = user.root; 6 = system; 22 = SAP-E01-D05)

In this example, component #3, which is SAP-D01-CI, has used 48.86 CPU seconds within the 15–minute interval and the memory consumption was 703455232 bytes.

- The `CPUSeconds` represent the utilized CPU seconds of the application component within the fixed measurement interval of 15 minutes. In the same way `CPUNormSeconds` represent the normalized utilization.
- `MemoryBytes` represent the utilized memory allocation of the application component, which is a snapshot at the end of the measurement interval.
- You always find an entry for the application component `Free`, that represents the unused resource capacity on the host (free memory, unused CPU seconds). In the case of memory, this value can be negative.
- You always find an entry for the application component `Other`, that represents the resource consumptions that could not be assigned to any SRM Project.
- The sum of all entries represents the servers capacity at the given point in time. For example, the sum of `MemoryBytes` give the available RAM capacity.

In addition to the MDX keywords, the following N1 AA Analyzer specific placeholders were introduced:

TABLE A–3 Analyzer Placeholders

Placeholder	Description
[$ServerGroup]	Constant member set of *ServerGroup* dimension containing a single member, equals the actual server group at runtime.
[$PhysicalHost]	Variable member set of *ServerGroup* dimension and *PhysicalHost* hierarchy, equals the selected set of physical hosts at runtime. Using this placeholder will enable the *Physical hosts* section in selection screen when executing an analysis.
[$Application]	Variable member set of Application dimension, equals the selected set of applications at runtime. Using this placeholder will enable the *Applications* section in selection screen when executing an analysis.
[$Component]	Variable member set of Application dimension and Component hierarchy, equals the selected set of components at runtime. Using this placeholder will enable the *Components* section in selection screen when executing an analysis.
[$Consumptions]	Variable member set as cross product of *ServerGroup* dimension and *PhysicalHost* hierarchy and Application dimension and Component hierarchy, equals the selected set of consumptions. Each happened on a dedicated physical host by a dedicated component. Using this placeholder will enable the `Dedicated consumptions` section in selection screen when executing an analysis.
[$TimeSlice_Minute]	Variable member set of *TimeSlice* dimension, equals the selected timeframe at runtime on a minute-based resolution. Using this placeholder will enable the *Timeframe* section in selection screen when executing an analysis. All time entries are represented in the UTC time zone when displaying analysis results in tables or diagrams.
[$TimeSlice_Hour]	Same as above, on a hour-based resolution (termed: aggregation).
[$TimeSlice_Day]	Same as above, on a day-based resolution (termed: aggregation).

TABLE A–3 Analyzer Placeholders *(Continued)*

Placeholder	Description
[$TimeSlice_Week]	Same as above, on a week-based resolution (termed: aggregation).
[$TimeSlice_Month]	Same as above, on a month-based resolution (termed: aggregation).
[$TimeSlice_Quarter]	Same as above, on a quarter-based resolution (termed: aggregation).
[$TimeSlice_Year]	Same as above, on a hour-based resolution (termed: aggregation).
[$TimeSlice]	Same as [$TimeSlice_Minute], but in dispose of raw times written by the Performance Collector daemon. **Note –** Do not use this placeholder. All time entries in analysis results will be 15 minutes late.
[$*constantname*]	Constant that is replaced with the value maintained in Analyzer Customizing. See "Basic Setup" on page 20. The value will be taken from parameter `none.analyzer.constants.`*constantname*. Within a dedicated server group, the value will be overwritten by parameter *servergroup*`.analyzer.constants.`*constantname* only if this parameter is maintained, which is optional.