Sybase, Inc.
One Sybase Drive
Dublin, CA 94568
www.sybase.com

Sybase Avaki EII Command Reference

Release 7.0

August 24, 2006

DOCUMENT ID: DC00577-01-0700-01

LAST REVISED: August 2006

Set in Arial, `Courier New`, and Times New Roman. Stanley Morison, the creator of Times New Roman, said of it: “By the vice of Mammon and the misery of the machine, it is bigoted and narrow, mean and puritan.”

Credits

This product includes software developed by the Apache Software Foundation (http://www. apache.org). This product includes Hypersonic SQL and ANTLR. This product includes code licenses from RSA Security, Inc. Some portions licensed from IBM are available at http://oss.software.ibm.com/icu4j/. Contains IBM® 64-bit Runtime Environment for AIX™, Java™ 2 Technology Edition Version 1.4 Modules © Copyright IBM Corporation 1999, 2000 All Rights Reserved. Contains the SAXON XSLT Processor from Michael Kay, which is available at http://saxon.sourceforge.net. This product includes software developed by the Proxool Project (http://proxool.sourceforge.net).

Sybase Avaki EII Command Reference
Written by Cheryl Magadieu and Beth Thoenen

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Table of contents

Preface

This *Sybase Avaki EII Command Reference* describes commands in the Sybase Avaki EII command line interface (CLI). The CLI is an alternative to Avaki Studio and to Avaki's web user interface. In addition, the CLI provides ways to perform certain actions that cannot be performed in Avaki Studio or through the web user interface.

Avaki supports the following types of commands:

- CLI commands. These commands all begin with "avaki" or are accessible within an Avaki shell, and are available on grid servers and command clients; and
- Avaki server start/stop commands.These commands do not begin with "avaki" and are not accessible within an Avaki shell. Start/stop commands are available on any machine that has an appropriate Avaki software installation.

Note This book and the product's user interfaces refer to Sybase Avaki EII software as *Avaki* or *Avaki Data Grid*.

Organization

This book is organized as follows:

Chapter 1 Using the CLI	Shows how to access the Avaki CLI and when to use it.
Chapter 2 Starting and stopping Avaki servers	Describes the commands for starting and stopping Avaki grid, DGAS, and proxy, share servers.
Chapter 3 Command reference	Explains how to use Avaki CLI commands.
Appendix A Attribute reference	Describes attributes you can set with the **avaki attribute** commands.
Appendix B Setting up DGAS properties files	Provides a sample properties file for data grid access servers.
Appendix C Configuring cron schedules	Describes the values that may be used to create a cron specification to indicate the schedule by which various tasks may be performed.
Appendix D Data type mapping files	Describes the format and contents of files you can use to map data types for provisioned SQL views.
Glossary	Defines terms used in this guide.

Related documentation and online help

Manuals

These manuals make up the Avaki documentation set:

- *Sybase Avaki EII Overture*
- *Sybase Avaki EII Administration Guide* (includes installation instructions)
- *Data Integration with Sybase Avaki Studio*
- *Sybase Avaki EII Provisioning and Advanced Data Integration Guide*
- *Sybase Avaki EII API Guide*
- *Sybase Avaki EII Command Reference*

The manuals are included, in PDF format, on the CD with the Avaki software. They are stored in the docs subdirectory of the Avaki installation directory.

To access the manuals via Avaki's web user interface, log in to your Avaki domain and click the **Help** link at the top right corner of any page of the web UI.

Online help

In addition to the manuals, Avaki provides online help for commands.

To display a list of Avaki commands with brief descriptions, log in to Avaki and enter **avaki help**:

```
% avaki help
List of domain commands:
        attribute
        backup
        cache
        cat
        categories
        cd
        chmod
        chown
        client
        cp
        dataservice
        dbconn
        dbop
        dgas
```

```
directory
domain
executionservice
file
group
help
id
ldap
ln
locks
login
logout
ls
mkdir
monitor
mv
nis
passwd
permissions
plugin
proxy
pwd
replica
rm
scheduleexclusion
search
security
server
share
shell
sqlview
status
systemproperty
upgrade
user
view
virtualdatabase
virtualschema
whoami
```

To display a description of a particular command and the syntax, enter a command of the form **avaki help <***command***>**. For example:

```
% avaki help mv
usage: avaki mv <source-grid-path> <target-grid-path>
    Description: Move or rename a grid directory or a file
    in a grid directory. Similar to the Unix mv command.
```

Conventions

This section describes text conventions used in this guide to represent elements of commands and screen displays.

Command syntax conventions

This table describes conventions that this book uses in command syntax statements. The "Enter this" column tells you whether you need to enter the characters when you type a command. The examples in the "Examples" column are not necessarily complete commands.

Convention	Description	Enter this?	Examples
`[ ]`	Square brackets surround optional arguments.	no	`avaki login [<user-id>]`
`{ }`	Curly brackets surround groups of required arguments.	no	`avaki chmod {--allow | --deny | --unset}`
`{ | }` `[ | ]`	Vertical bars separate alternative options within square or curly brackets. If the brackets are square, you need not enter any of the options; if the brackets are curly, you must choose one of the options.	no	`avaki backup {--snapshot | --recover}` `avaki share --create [--background | --bg]`
`< >`	Angle brackets surround placeholder arguments that you must replace with a value such as a path or file name. Square brackets outside the angle brackets indicate that the placeholder is optional.	no	`avaki help [<command-name>]` `avaki help share`
`*`	An asterisk follows an argument that can be entered zero or more times.	no	`avaki plugin --generate [--input=<stream-spec>*]` `avaki plugin --generate --input="name=input1;type=XML"`
`+`	A plus sign follows an argument that can be entered one or more times. Use spaces to separate the values.	no	`avaki cat <grid-path>+` `avaki cat /home/fred/file1 /home/fred/file2`

Convention	Description	Enter this?	Examples
-	Enter a hyphen or minus sign before a single-letter command option.	yes	`avaki mkdir -p`
=	Enter an equal sign before the value of an option.	yes	`avaki login --auth-service=<auth-service-name>`
(space)	A space separates multiple arguments.	yes	`avaki cat file1 file2 file3`

Conventions for screen examples

This table describes conventions this book uses in examples of user input and system output.

Convention	Description	Example
`$ or C:>`	The command prompt	`$`
`<>`	A placeholder; replace the text inside the brackets with an option or value	`$ avaki ls <grid-path>`
`screen font`	Text that appears on the screen	`sample text`
`bold screen font`	User input—commands that you enter	`$ avaki ls`

How to contact Avaki support at Sybase, Inc.

For general information about Sybase technical support, see the *Customer Service Reference Guide* at

http://www.sybase.com/support/aboutsupport/guide/csrg

Please contact us with any questions or difficulties you encounter.

By telephone

In North America, call toll free: 1-800-8SYBASE

Outside North America, follow the link below to see a list of Sybase offices and phone numbers around the world.

http://www.sybase.com/contactus/support

On the web

If you are a designated contact for a technical support plan, you can log and track cases on the web using the Case Express application. At **http://www.sybase.com**, mouse over the **Support and Services** tab and select **Case Management** from the drop-down list. Use the email address and password for your mysybase account to log in.

Chapter 1

Using the CLI

This chapter discusses when the Avaki command line interface (CLI) is useful and how to access it. The chapter is organized as follows:

Before using the CLI

This *Sybase Avaki EII Command Reference* assumes that you know about the components of Avaki data grids. These include database tools such as data services, database operations, and SQL views; Avaki's access control and caching schemes; and the servers that make up an Avaki domain—grid domain controllers, grid servers, share servers, etc. If you're not familiar with Avaki data grids, read the *Sybase Avaki EII Overture* before you use the commands.

When to use the CLI

Most Avaki tasks are easiest to perform using Avaki Studio or the web user interface. However, you can often save time on frequently performed, repetitious, or complex tasks if you use the CLI for scripts.

Accessing the CLI

The Avaki CLI is installed automatically when grid servers (including grid domain controllers) are installed. You can install the Avaki CLI on any Avaki server by running the installer and choosing to install the command line client feature.

Note The start and stop commands for Avaki servers are not considered part of the CLI and are, of course, available on the servers. For information on these commands, see Chapter 2, "Starting and stopping Avaki servers".

Before you can use the CLI, Avaki software must be installed and started on at least one machine, your grid domain controller (GDC). If you plan to use the CLI on a grid server or a command-line client, an appropriate installation must be performed on the machine you plan to use. See the *Sybase Avaki EII Administration Guide* for installation instructions, or ask your Avaki domain administrator for assistance.

Some of the commands in this command reference pertain to administrative tasks that may be performed only by users who have administrative privileges.

In the Avaki CLI, relative paths are relative to the current working directory of the client session. To set the current working directory, use the DOS command **CD** or the Unix command **cd**.

Before using Avaki commands on any machine (including a machine running a GDC), follow these steps to connect to a GDC:

Step 1 Open a command window and enter a command of this form:

```
avaki client --connect <GDC-machine>
```

<GDC-machine> is the DNS name or IP address of a machine on which an Avaki GDC is installed and operating. For example:

```
$ avaki client --connect Flintstone
```

Step 2 Log in to the domain. In this example, the user enters a password at the Password prompt, and at the next two prompts presses **Enter** to accept the defaults (Default-AuthService and Grid service type):

```
$ avaki login Fred
Password:
Auth service [DefaultAuthService]:
Service type [(Grid), Nis, Ldap]:
Fred logged in successfully
$
```

Chapter 2

Starting and stopping Avaki servers

This chapter describes the commands for starting and stopping Avaki grid, DGAS, proxy, and share servers. The commands are listed alphabetically.

The commands described in this chapter must be entered in the Avaki installation directory (on Windows, the default is C:\AvakiDataGrid*xx*, where *xx* represents the release number).

Automatically restarting servers and GDCs

Avaki provides an automatic restart feature for Avaki servers and grid domain controllers. When auto-restart is enabled, the Avaki server or GDC restarts automatically when the computer restarts. If auto-restart is not enabled, you must bring the GDC or server back up manually each time its machine reboots. If you plan to use Avaki's auto-restart feature in Unix, you must be logged in to the machine as a user who has administrative privileges when you set up the GDC or server. If you plan to use Avaki's auto-restart feature in Windows, you must be logged in to the machine as a user who has permission to log on as a service. In Windows, when you start a server with auto-restart enabled, the server is registered as a service. You can find an entry for the service in the machine's Services list (Start > Control Panel > Administrative Tools > Services). In Unix, when you start a server with auto-restart enabled, the run command script for the server will be placed in the machine's /etc/rc/* directory.

dgas --register

Start a data grid access server with the automatic restart feature enabled.

Note In Windows, you can use Windows shortcuts to start Avaki servers with auto-restart enabled. To start a data grid access server with auto-restart enabled, open the Start menu, then select the **Avaki Data Grid** program group. Select **DGAS**, and then select the **Register Data Grid Access Server as Windows service** shortcut.

The server takes a minute or two to start.

Syntax

```
dgas --register [--user=<user>] [--name=<server-name>]
   [--db-path=<local-path>] [--cache-path=<local-path>]
   [--port=<connect-port>] [--rmi-port=<rmi-server-port>]
```

`--user=<user>`	Optional. The name of the nonadministrative user account that you set up to run Avaki services. The account must have logon-as-service privileges. For details about configuring user accounts, see the *Sybase Avaki EII Administration Guide*.
`--name=<server-name>`	Optional. Specify a name for the data grid access server. If you don't specify a name for a new DGAS, the system uses the default name Access1.
`--db-path=<local-path>`	Optional. Specify the location of the DGAS state database, which stores property settings and other information required to restart the DGAS. Default: <Avaki-install-directory>/DGAS/dgas_db
`--cache-path=` `<local-path>`	Optional. Specify the location of the local directory that holds all DGAS internal caches. Default: <Avaki-install-directory>/DGAS/cache

`--port=<connect-port>`	Optional. Specify the connect port that the DGAS will use to communicate with other objects in the data grid. Default value: 1399
`--rmi-port=<rmi-server-port>`	Optional. Specify the RMI server port that the DGAS will use. Default value: 1599

dgas --start

Start a data grid access server (DGAS) without enabling the automatic restart feature. For information about enabling auto-restart, see **dgas --register** on page 6.

The DGAS takes a minute or two to start.

Syntax

```
dgas --start [--name=<server-name>] [--db-path=<local-path>]
  [--cache-path=<local-path>] [--new] [--port=<connect-port>]
  [--rmi-port=<rmi-server-port>]
```

`--name=<server-name>`	Optional. Specify a name for the data grid access server. If you don't specify a name for a new DGAS, the system uses the default name.
`--db-path=<local-path>`	Optional. Specify the location of the DGAS state database, which stores property settings and other information required to restart the DGAS. Default: <Avaki-install-directory>/DGAS/dgas_db
`--cache-path=<local-path>`	Optional. Specify the location of the local directory that holds all DGAS internal caches. Default: <Avaki-install-directory>/DGAS/cache

`--new`	Optional. Wipe out the state database for the previous DGAS but retain any DGAS internal caches.
`--port=<connect-port>`	Optional. Specify the connect port that the DGAS will use to communicate with other objects in the data grid. Default value: 1399
`--rmi-port=<rmi-server-port>`	Optional. Specify the RMI server port that the DGAS will use. Default value: 1599

dgas --stop

Stop a data grid access server that is running on Unix. For information about disabling auto-restart, see **dgas --unregister** on page 9.

Note The procedure for stopping a data grid access server on Windows depends on whether auto-restart is enabled for the server, as follows:

- If auto-restart is not enabled on the server, enter **Ctrl-C** in the server's window to stop it.
- If auto-restart is enabled on the server, use the Windows Services list to stop and restart the server. Right-click on the server name (for example, Avaki-DGAS-1096608202) and select **stop** or **restart**.

Syntax

```
dgas --stop [--name=<server-name>]
```

`--name=<server-name>`	Optional. Specify a name for the data grid access server. If you don't specify a name for a new DGAS, the system uses the default name.

dgas --unregister

Stop a data grid access server and disable the automatic restart feature.

Note In Windows, you can use Windows shortcuts to stop Avaki servers and disable auto-restart. To stop a data grid access server and disable auto-restart, open the Start menu, then select the **Avaki Data Grid** program group. Select **DGAS**, and then select the **Unregister Data Grid Access Server as Windows service** shortcut.

Syntax

```
dgas --unregister [--name=<server-name>]
```

`--name=<server-name>`	Optional. Specify a name for the data grid access server. If you don't specify a name for a new DGAS, the system uses the default name.

grid-server --register

Start a grid server or grid domain controller with the automatic restart feature enabled.

Note In Windows, you can use Windows shortcuts to start Avaki servers with auto-restart enabled. To start a grid server or grid domain controller with auto-restart enabled, open the Start menu, then select the **Avaki Data Grid** program group. Select **Grid Server**, and then select the **Register Grid Server as Windows service** shortcut.

The grid server takes a minute or two to start. Startup is complete when you see a message similar to one of these:

On Unix:

```
bash-2.05# ./grid-server --register
User to run as [wilma]:
The Avaki-GridService1395600136 service is starting...
Starting Avaki-GridService1395600136 daemon:
```

```
The Avaki-GridService1395600136 service started. For more
details, see
/home/local/wilma/AvakiDataGrid70/jboss/server/grid-server/
log/server.log
```

On Windows:

```
C:\AvakiDataGrid70> grid-server --register
User to run as [BEDROCK\wilma] :
Password to run as BEDROCK\wilma:
The Avaki-GridService-1096607241 service is starting.
The Avaki-GridService-1096607241 service was started
successfully.
```

Syntax

```
grid-server --register [--user=<user> [--password=<password>]]
```

`--user=<user>`	Optional. The name of the nonadministrative user account that you set up to run Avaki services. The account must have logon-as-service privileges. For details about configuring user accounts, see the *Sybase Avaki EII Administration Guide*.
`--password=<password>`	Optional. The password for the user account specified in <user>.

grid-server --start

Start a grid server or grid domain controller without enabling the automatic restart feature. For information about enabling auto-restart, see **grid-server --register** on page 9.

The grid server takes a minute or two to start. Startup is complete when you see a message similar to this:

```
2003-06-30 13:50:26,951 INFO
[org.jboss.system.server.Server, main] JBoss
(MX MicroKernel) [3.0.6 (CVSTag=JBoss_3_0_6
Date=200301260037)] Started in 1m:3s:212ms
```

Syntax

```
grid-server --start
```

–	This command has no options.

grid-server --stop

Stop a grid server or grid domain controller that is running on Unix. For information about disabling auto-restart, see **grid-server --unregister** on page 12.

Note The procedure for stopping a grid server or grid domain controller on Windows depends on whether auto-restart is enabled for the server, as follows:

- If auto-restart is not enabled on the server, enter **Ctrl-C** in the server's window to stop it.
- If auto-restart is enabled on the server, use the Windows Services list to stop and restart the server. Right-click on the server name (for example, Avaki-GridService-1096608202) and select **stop** or **restart**.

Syntax

```
grid-server --stop
```

–	This command has no options.

grid-server --unregister

Stop a grid server or grid domain controller and disable the automatic restart feature.

Note In Windows, you can use Windows shortcuts to stop Avaki servers and disable auto-restart. To stop a grid server or grid domain controller and disable auto-restart, open the Start menu, then select the **Avaki Data Grid** program group. Select **Grid Server**, and then select the **Unregister Grid Server as Windows service** shortcut.

Syntax

```
grid-server --unregister
```

–	This command has no options.

proxy-server --register

Start a proxy server with the automatic restart feature enabled.

Note In Windows, you can use Windows shortcuts to start Avaki servers with auto-restart enabled. To start a proxy server with auto-restart enabled, open the Start menu, then select the **Avaki Data Grid** program group. Select **Proxy Server**, and then select the **Register ProxyServer as Windows service** shortcut.

The proxy server takes a minute or two to start.

Syntax

```
proxy-server --register [--user=<user> [--password=<password>]]
```

`--user=<user>`	Optional. The name of the nonadministrative user account that you set up to run Avaki services. The account must have logon-as-service privileges. For details about configuring user accounts, see the *Sybase Avaki EII Administration Guide*.
`--password=<password>`	Optional. The password for the user account specified in <user>.

proxy-server --start

Start a proxy server without enabling the automatic restart feature. For information about enabling auto-restart, see **proxy-server --register** on page 13.

The proxy server takes a minute or two to start.

Syntax

```
proxy-server --start
```

–	This command has no options.

proxy-server --stop

Stop a proxy server that is running on Unix. For information about disabling auto-restart, see **proxy-server --unregister** on page 15.

Note The procedure for stopping a proxy server on Windows depends on whether auto-restart is enabled for the server, as follows:

- If auto-restart is not enabled on the server, enter **Ctrl-C** in the server's window to stop it.
- If auto-restart is enabled on the server, use the Windows Services list to stop and restart the server. Right-click on the server name (for example, Avaki-ProxyService-1096608202) and select **stop** or **restart**.

Syntax

```
proxy-server --stop
```

–	This command has no options.

proxy-server --unregister

Stop a proxy server and disable the automatic restart feature.

Note In Windows, you can use Windows shortcuts to stop Avaki servers and disable auto-restart. To stop a proxy server and disable auto-restart, open the Start menu, then select the **Avaki Data Grid** program group. Select **Proxy Server**, and then select the **Unregister ProxyServer as Windows service** shortcut.

Syntax

```
proxy-server --unregister
```

–	This command has no options.

share-server --register

Start a share server with the automatic restart feature enabled.

Note In Windows, you can use Windows shortcuts to start Avaki servers with auto-restart enabled. To start a share server with auto-restart enabled, open the Start menu, then select the **Avaki Data Grid** program group. Select **Share Server**, and then select the **Register ShareServer as Windows service** shortcut.

The share server takes a minute or two to start.

Syntax

```
share-server --register [--user=<user> [--password=<password>]]
```

`--user=<user>`	Optional. The name of the nonadministrative user account that you set up to run Avaki services. The account must have logon-as-service privileges. For details about configuring user accounts, see the *Sybase Avaki EII Administration Guide*.
`--password=<password>`	Optional. The password for the user account specified in <user>.

share-server --start

Start a share server without enabling the automatic restart feature. For information about enabling auto-restart, see **share-server --register** on page 16.

The share server takes a minute or two to start.

Syntax

```
share-server --start
```

–	This command has no options.

share-server --stop

Stop a share server that is running on Unix. For information about disabling auto-restart, see **share-server --unregister** on page 18.

Note The procedure for stopping a share server on Windows depends on whether auto-restart is enabled for the server, as follows:

- If auto-restart is not enabled on the server, enter **Ctrl-C** in the server's window to stop it.
- If auto-restart is enabled on the server, use the Windows Services list to stop and restart the server. Right-click on the server name (for example, Avaki-ShareService-1096608202) and select **stop** or **restart**.

Syntax

```
share-server --stop
```

–	This command has no options.

share-server --unregister

Stop a share server and disable the automatic restart feature.

Note In Windows, you can use Windows shortcuts to stop Avaki servers and disable auto-restart. To stop a share server and disable auto-restart, open the Start menu, then select the **Avaki Data Grid** program group. Select **Share Server**, and then select the **Unregister ShareServer as Windows service** shortcut.

Syntax

```
share-server --unregister
```

– This command has no options.

Chapter 3

Command reference

This chapter describes Avaki CLI commands. The commands are listed alphabetically.

avaki attribute --delete

Delete an attribute for a grid file, directory, Avaki share, cache, server, or service.

Syntax

```
attribute {--delete | -d} [-r] --name=<attribute-name>
  --type={string | integer | float | date | time | timestamp}
  <target-grid-path>
```

`-r`	Recursive mode (optional). If you are deleting an attribute for a directory, specify the -r option if you want to delete the attribute recursively from all the directory's contents. If you do not specify the -r option, the attribute will be deleted only from the directory itself.
`--name= <attribute-name>`	The name of the attribute to delete.

`--type={string | integer | float | date | time | timestamp}`

The attribute's type, which is one of the following:

string	Text, such as search strings or a project name.
integer	Any whole quantity, such as the maximum size of the cache on a machine.
float	A numeric value that can be fractional or very large.
date	Year, month, and date. Format: yyyy-mm-dd Example: 2003-09-21
time	Hour, minute, and second that an event occurs. Format: hh:mm:ss Example: 23:59:59.
time-stamp	A precise time and date, including year, month, date, hour, minute, second, and (optionally) fraction of a second. Format: yyyy-mm-dd hh:mm:ss[.$\overline{ff}$] There is a space between the day and the hour. Example: 2003-09-21 04:05:06.57567

`<target-grid-path>`

The data catalog path to the file, directory, or other object for which you are deleting an attribute.

avaki attribute --list

Display the name, type, and value for all of an object's attributes.

Syntax

```
attribute {--list | -l} [-r] <target-grid-path>
```

`-r`	Recursive mode (optional). If you are obtaining a list of attributes for a directory, specify the -r option if you want information about the attributes for all of the directory's contents. If you do not specify the -r option, the output will include information about only the directory itself.
`<target-grid-path>`	The relative or absolute data catalog directory for the file, directory, or other object for which you are adding or modifying an attribute.

avaki attribute --update

Add or modify an attribute for a grid file, directory, Avaki share, cache, server, or service. See Appendix A, "Attribute reference", for a list of settable attributes.

Syntax

```
avaki attribute {--update | -u} [-r] --name=<attribute-name>
  --type={string | integer | float | date | time | timestamp}
  --value=<attribute-value> <target-grid-path>
```

`-r`	Recursive mode (optional). If you are configuring an attribute for a directory, specify the -r option if you want the attribute's settings to apply recursively to all the directory's contents. If you do not specify the -r option, the settings apply only to the directory itself.
`--name=<attribute-name>`	The name of the attribute. The name may be up to 255 characters long, and it must not include spaces or special characters (such as a backslash, an ampersand, or an exclamation point).

`--type={string | integer | float | date | time | timestamp}`

The attribute's type, which is one of the following:

string	Text, such as search strings or a project name. The string may be up to 255 characters long.
integer	Any whole quantity, such as the maximum size of the cache on a machine. The valid range for integers is -9,223,372,036,854,775,808 to +9,223,372,036,854,775,807.
float	A numeric value that can be fractional or very large. This might be the load on a computer or a percentage, for example. The valid range for float is ±4.94065645841246544e-324 to ±1.79769313486231570e+308.
date	Year, month, and date, such as a project date, file creation date, or a subject's birth date. Format: yyyy-mm-dd Example: 2003-09-21
time	Hour, minute, and second that an event occurs. This can be a celestial event or flight schedule information, for example. The supported range is 00:00:00 to 23:59:59. Format: hh:mm:ss Example: 04:05:06
timestamp	A precise time and date, including year, month, date, hour, minute, second, and (optionally) fraction of a second. This attribute type can be used to mark the start and end of an experiment or a chemical reaction, for example. Format: yyyy-mm-dd hh:mm:ss[.ff] There is a space between the day and the hour. Example: 2003-09-21 04:05:06.57567

`--value=<attribute-value>`	The value you're assigning to the attribute specified in the --name option. The formats for the date, time, and timestamp attribute types are as follows: **date** — Year, month, and date; format: yyyy-mm-dd Example: 2003-09-21 **time** — Hour, minute, and second; format: hh:mm:ss Example: 04:05:06 **timestamp** — A precise time and date, including year, month, date, hour, minute, second, and (optionally) fraction of a second. Format: yyyy-mm-dd hh:mm:ss[.ff] There is a space between the day and the hour. Example: 2003-09-21 04:05:06.57567
`<target-grid-path>`	The data catalog directory of the file, directory, database operation, cache service, or other object for which you are adding or modifying an attribute.

Example

This command creates a user-defined string attribute called SlateRockDocs for the share named DocProjects and sets the value to ReleaseNotes.

```
$ avaki attribute -u --name=SlateRockDocs --type=string
--value=ReleaseNotes /Shares/DocProjects
```

avaki backup

Back up or recover a grid server's persistent state information, including databases and internal shares. (The databases store state information about the grid domain, such as grid authentication services and access control lists.) We recommend running the backup command only when the grid is not under load.

Syntax

```
avaki backup {--snapshot | --recover}
  [--grid-server=<grid-server>] <filename>.zip
```

`--snapshot`	Create a snapshot of the database.
`--recover`	Reset database state to a previously taken snapshot.
`--grid-server=<grid-server>`	Optional. Specify the name of the Avaki server to back up. For example, Bedrock. If you omit this option, the GDC will be backed up.
`<filename>.zip`	The name of the local file that you are saving a snapshot to or recovering a backup from. You can specify an absolute path or a relative path.

avaki cache --evict

Purge one or more database operations, data services, files, or directories from the cache service and unschedule them if they're pinned.

To use this command, you must have write permission on the cache service (members of the Administrators group and the DataProviders group have this permission by default).

If you want to invalidate an item in the cache rather than permanently evicting it, see the invalidation commands, beginning with **avaki cache --invalidate** (page 28).

Syntax

```
avaki cache --evict <cache-service-path>
  [--dbop=<qualified-dbop-name>* |
  --data-service=<qualified-ds-name>*] [-r] <grid-path>*
```

`<cache-service-path>`	The data catalog path of a cache service.
`--dbop=<qualified-dbop-name>*`	Optional. The names of one or more database operations, in this format: [<domain-name>.]<database-connector-name>.<database-operation-name>
`--data-service=<qualified-ds-name>*`	Optional. The names of one or more data services, in this format: `[<domain-name>.]<data-service-name>`
`-r`	Optional. If an object specified in <grid-path> is a directory, recursively evict the directory and all its contents from the cache.
`<grid-path>*`	Optional. The data catalog paths of one or more files or directories.

avaki cache --evict --all

Purge all database operations, data services, files, or directories from the cache service and unschedule them if they're pinned.

To use this command, you must have write permission on the cache service (members of the Administrators group and the DataProviders group have this permission by default).

If you want to invalidate an item in the cache rather than permanently evicting it, see the invalidation commands, beginning with **avaki cache --invalidate** (page 28).

Syntax

```
avaki cache --evict --all <cache-service-path>
```

`<cache-service-path>`	The grid path of a cache service.

avaki cache --evict --deleted

Purge from the cache any items whose source objects (database operations, data services, files, or directories) have been deleted from the domain. The cache service attempts to refresh a cache entry whenever the item's source objects have been deleted, but users may want to use **avaki cache --evict --deleted** to expedite the process in order to save disk space.

To use this command, you must have write permission on the cache service (members of the Administrators group and the DataProviders group have this permission by default).

If you want to invalidate an item in the cache rather than permanently evicting it, see the invalidation commands, beginning with **avaki cache --invalidate** (page 28).

Syntax

```
avaki cache --evict -deleted [--dbops | --data-services |
  --filesystem-objects] <cache-service-path>
```

`--dbops`	Optional. Purge from the cache all database operations that have been deleted from the grid.
`--data-service`	Optional. Purge from the cache all data services that have been deleted from the grid.
`--filesystem-objects`	Optional. Purge from the cache all files or directories that have been deleted from the grid.
`<cache-service-path>`	The grid path of a cache service.

avaki cache --get

Display the name of the cache service that is associated with the specified DGAS, grid server, or local user. The association is valid across avaki logins to the same CLI.

To use this command, you must have read permission on the cache service (members of the Administrators group and the DataProviders group have this permission by default) and on the DGAS, grid server, or local user.

Syntax

```
avaki cache --get {<DGAS> | <grid-server> | --cli}
```

`<DGAS>`	The data catalog path of a data grid access server.
`<grid-server>`	The data catalog path of a grid server.
`--cli`	Indicates the current local user's instance of the CLI.

avaki cache --invalidate

Flag one or more database operations, data services, files, or directories in a cache as invalid. When a cached copy of an object is invalidated, the cached database operation metadata is invalidated and the current cached data will no longer be used when a user requests the object. The object will be refreshed in the cache the next time it is accessed or upon its next scheduled refresh.

To use this command, you must have write permission on the cache service (members of the Administrators group and the DataProviders group have this permission by default).

Syntax

```
avaki cache --invalidate <cache-service-path>
  --dbop=<qualified-dbop-name>*
  --data-service=<qualified-ds-name>* [-r] <grid-path>*
```

`<cache-service-path>`	The data catalog path of a cache service.
`--dbop=qualified-dbop-name*`	Optional. The names of one or more database operations, in this format: `[<domain-name>.]<database-connector-name>.<database-operation-name>`
`--data-service=qualified-ds-name*`	Optional. The names of one or more data services, in this format: `[<domain-name>.]<data-service-name>`
`-r`	Optional. If an object specified in <grid-path> is a directory, recursively invalidate the directory and all its contents.
`<grid-path>*`	Optional. The data catalog paths to one or more files or directories.

avaki cache --invalidate --all

Flag all database operations, data services, files, or directories in a cache as invalid. When a cached copy of an object is invalidated, the cached copy data will no longer be used when a user requests the object; the object will be refreshed in the cache the next time it is accessed or upon its next scheduled refresh.

To use this command, you must have write permission on the cache service (members of the Administrators group and the DataProviders group have this permission by default).

Syntax

```
avaki cache --invalidate --all <cache-service-path>
```

`<cache-service-path>`	The grid path of a cache service.

avaki cache --invalidate-dataservice-results

Invalidate the results from a cached data service that correspond to the specified input parameters, but do not invalidate cached data service metadata. When a cached copy of an object is invalidated, the current cached data will no longer be used when a user requests the object; the object will be refreshed in the cache the next time it is accessed or upon its next scheduled refresh.

To use this command, you must have write permission on the cache service (members of the Administrators group and the DataProviders group have this permission by default).

Syntax

```
avaki cache --invalidate-dataservice-results
  <cache-service-path> <qualified-ds-name>
  [<param-name>:<param-value>]+
```

`<cache-service-path>`	The data catalog path of a cache service.
`qualified-ds-name`	Optional. The name of a data service, in this format: `[<domain-name>.]<data-service-name>`
`[<param-name>:<param-value>]+`	Optional. Specify one or more input or output parameters and associated values required by the data service.

avaki cache --invalidate-dbop-results

Invalidate the results from a cached database operation that correspond to the specified input parameters, but do not invalidate cached database operation metadata. When a cached copy of an object is invalidated, the cached data will no longer be used when a user requests the object; the object will be refreshed in the cache the next time it is accessed or upon its next scheduled refresh.

To use this command, you must have write permission on the cache service (members of the Administrators group and the DataProviders group have this permission by default).

Syntax

```
avaki cache --invalidate-dbop-results <cache-service-path>
  <qualified-dbop-name> [<parameter-value>]+
```

`<cache-service-path>`	The grid path of a cache service.
`qualified-dbop-name`	The names of a database operation, in this format: `[<domain-name>.]<database-connector-name>.<database-operation-name>`
`<parameter-value>+`	Optional. One or more parameter values required by the target database operation. The parameter values will be processed in the order specified in the database operation's query. Any parameter value that includes a space must be enclosed in quotation marks. Any string at the end of the command that is enclosed in quotation marks will be interpreted as a single value. (That is, enclose each parameter value in its own pair of quotation marks.)

avaki cache --list

Display the tagging information for the given cached objects (files, directories, database operations, and data services) contained in the specified cache. If no objects are specified for the given cache service, the tagging information for the entire collection of cached objects is returned.

To use this command, you must have read permission on the cache service.

Syntax

```
avaki cache --list <cache-service-path>
   [--dbop=<qualified-dbop-name>]*
   [--data-service=<qualified-ds-name>]* [-r] [<grid-path>*]
```

`<cache-service-path>`	The data catalog path of a cache service in the current Avaki domain.
`--dbop=<qualified-dbop-name>*`	Optional. Specify the name of a cached database operation, in this format: `[<domain-name>.]<database-connector-name>.<database-operation-name>`
`--data-service=<qualified-ds-name>*`	Optional. Specify the name of a cached data service, in this format: `[<domain-name>.]<data-service-name>`
`-r`	Optional. If an object specified in <grid-path> is a directory, recursively list the tagging information for the directory and any cached objects within its hierarchy.
`<grid-path>*`	Optional. Specify the data catalog path of a cached grid file or directory.

avaki cache --refresh

Force the cache service to inspect (and update, if necessary) the specified cached entry.

To use this command, you must have write permission on the cache service (members of the Administrators group and the DataProviders group have this permission by default) and on the target files or directories.

Syntax

```
avaki cache --refresh <cache-service-path>
  [--dbop=<qualified-dbop-name>]*
  [--data-service=<qualified-ds-name>]* [-r] [<grid-path>*]
```

`<cache-service-path>`	The data catalog path of a cache service.
`--dbop=<qualified-dbop-name>*`	Optional. The names of one or more database operations, in this format: `[<domain-name>.]<database-connector-name>.<database-operation-name>`
`--data-service=qualified-ds-name*`	Optional. The names of one or more data services, in this format: `[<domain-name>.]<data-service-name>`
`-r`	Optional. If an object specified in <grid-path> is a directory, recursively refresh the directory and all its contents in the cache.
`<grid-path>*`	The data catalog paths of one or more files or directories.

avaki cache --set

Configure the specified client to use a cache service.

To use this command, you must have read permission on the cache service (members of the Administrators group and the DataProviders group have this permission by default) and on the DGAS, grid server, or local user.

Syntax

```
avaki cache --set {<DGAS> | <grid-server> | --cli}
  <cache-service-path>
```

`<DGAS>`	The data catalog path of a data grid access server.
`<grid-server>`	The data catalog path of a grid server.
`--cli`	Indicates the current local user's instance of the CLI.
`<cache-service-path>`	The data catalog path of a cache service.

avaki cache --unset

Uncouple any configured cache service from the specified client. If you uncouple a grid server from a cache service, the grid server uses its default setting, in which it is not associated with a cache service.

To use this command, you must have read permission on the cache service (members of the Administrators group and the DataProviders group have this permission by default) and on the DGAS, grid server, or local user.

Syntax

```
avaki cache --unset {<DGAS> | <grid-server> | --cli}
```

`<DGAS>`	The grid path of a data grid access server.
`<grid-server>`	The grid path of a grid server.
`--cli`	Indicates the current local user's instance of the CLI.

avaki cat

Display the specified Avaki file. Similar to the Unix **cat** command.

Syntax

```
avaki cat <grid-path>+
```

`<grid-path>+`	The path to the file or files to be displayed. The file(s) must be in a grid directory. You can enter an absolute path or a relative one.

avaki categories --add-to-category

Add objects to a category.

Syntax

```
avaki categories --add-to-category <category-path>
  [--dbop=<qualified-dbop-name>]*
  [--data-service=<qualified-ds-name>]* [<grid-path>]*
```

`<category-path>`	The name of the category to which objects will be added, in this format:
	`/ParentCategoryName`
	To add objects to a subcategory, place a slash between the parent category name and the subcategory. For example:
	`/ParentCategoryName/SubcategoryName`

`<qualified-dbop-name>*`	The name of zero or more database operations, in this format: `[<domain-name>.]<database-connector-name>.<database-operation-name>`
`<qualified-ds-name>*`	The name of zero or more data services to add to the category, in this format: `[<domain-name>.]<data-service-name>`
`<grid-path>*`	Zero or more data catalog paths of objects to add to the category.

avaki categories --create

Create a category.

Syntax

```
avaki categories --create [--description=<description>] [-D |
  --domain=<domain-name>] <category-path>
```

`--description=<description-text>`	Optional. Enter details about this category.
`[-D | --domain=<domain-name>]`	Optional. The name of the domain in which to create the category.

`<category-path>`	The path of the category to create, in this format:
	`/ParentCategoryName`
	To create a subcategory, place a slash between the parent category name and the subcategory. For example:
	`/ParentCategoryName/SubcategoryName`

avaki categories --delete

Deletes the category <category-path>.

Syntax

```
avaki categories [-d | --delete] [-D | --domain=<domain-name>]
  <category-path>
```

`[-D | --domain=<domain-name>]`	Optional. The name of the domain from which the category will be deleted.
`<category-path>`	The path of the category to delete.

avaki categories --describe

Show the description for a category.

Syntax

```
avaki categories --describe [-D | --domain=<domain-name>]
  <category-path>
```

`[-D | --domain=<domain-name>]`	Optional. The name of the category's domain.
`<category-path>`	The path of the category to describe.

avaki categories --list

List the categories that exist in a particular domain.

Syntax

```
avaki categories --list [-r | --recursive] [-D |
  --domain=<domain-name>] [<category-path>]
```

`[-r | --recursive]`	Recursive mode (optional). Use the -r option with a blank category path to display all categories and subcategories for the domain.
`[-D | --domain=<domain-name>]`	Optional. The name of the domain for which a list of categories will be obtained.
`<category-path>`	Optional. List the subcategories of <category-path>.

avaki categories --remove-from-category

Remove objects from a category.

Syntax

```
avaki categories --remove-from-category [-D |
  --domain=<domain-name>]<category-path>
  [--dbop=<qualified-dbop-name>]*
  [--data-service=<qualified-ds-name>]* [<grid-path>*]
```

`[-D | --domain=<domain-name>]`	Optional. The name of the domain from which the category will be removed.
`<category-path>`	The path of the category from which the object will be removed.
`<qualified-dbop-name>*`	The name of zero or more database operations, in this format: `[<domain-name>.]<database-connector-name>.<database-operation-name>`
`<qualified-ds-name>*`	The name of zero or more data services, in this format: `[<domain-name>.]<data-service-name>`
`<grid-path>*`	Zero or more grid directories of objects that you are removing from a category.

avaki categories --set-description

Set the description for a category.

Syntax

```
avaki categories --set-description [-D |
  --domain=<domain-name>] <category-path> <description-text>
```

`[-D | --domain=<domain-name>]`	Optional. The name of the domain for which the category description will be set.
`<category-path>`	The path of the category for which the description will be set.
`--description=<description-text>`	Enter details about this category; enclose the entire text string in quotation marks. Length limit: 255 characters

avaki categories --show categories

Shows the categories that an object is a member of.

Syntax

```
avaki categories --show-categories
  [--dbop=<qualified-dbop-name> |
  --data-service=<qualified-ds-name> | <grid-path>]
```

`<qualified-dbop-name>`	The name of a database operation, in this format: `[<domain-name>.]<database-connector-name>.<database-operation-name>`
`<qualified-ds-name>`	The name of a data service, in this format: `[<domain-name>.]<data-service-name>`
`<grid-path>`	The data catalog directory of the object for which you want to obtain a list of categories.

avaki cd

Change the current working directory in a grid directory. Similar to the Unix and DOS **cd** commands.

Syntax

```
avaki cd <grid-path>
```

`<grid-path>`	The data catalog directory to which you are changing.

avaki chmod

Modify read, write, execute, and delete permissions for users and groups in the access control list for an Avaki object.

Syntax

```
avaki chmod {<qualified-user-name> |
  --user=<qualified-user-name> | <qualified-group-name> |
  --group=<qualified-group-name> | everyone |
  --filename=<interconnect-ID-file>}
  {--delete | {--allow | --deny | --unset}=<permission-name>}+
  {--target=<target> | <target>} [-r | --recursive] [--lock]
```

`<qualified-user-name> | --user=<qualified-user-name> | <group-grid-path> | --group=<group-grid-path> | <qualified-group-name> | --group=<qualified-group-name> | everyone | --filename=<interconnect-ID-file>`

Specify the qualified name of the users or groups for which permissions will be changed.

The format for the qualified user name is as follows:

`<user>@<authservice>.<authservice-type>.<domain>`

For example:

`wilma@DefaultAuthService.Grid.Bedrock`

The elements of <qualified-user-name> are as follows:

`<user>`	The name of the user to whose permissions you are modifying.
`<authservice>`	The name of the user's authentication service.
`<authservice-type>`	The type of the user's authentication service (Grid, Nis, or Ldap).
`<domain>`	The name of the user's Avaki domain.

The format for the qualified group name is as follows:

```
<group>@<authservice>.<authservice-type>.<domain>
```

For example:

```
Development@DefaultAthService.Grid.Bedrock
```

The elements of <qualified-group-name> are as follows:

`<group>`	The name of the group whose permissions you are modifying.
`<authservice>`	The name of the group's authentication service.
`<authservice-type>`	The type of the group's authentication service (Grid, Nis, or Ldap).
`<domain>`	The name of the group's Avaki domain.

Specify everyone if you want to change the permissions for all known and unknown users and groups.

`--filename=<interconnect-ID-file>`

Specify the path to an interconnect ID file for the user or group whose permissions you want to add or change. This option lets you edit permissions for a user or group in another grid domain. The file is assumed to be in the local file system.

`--delete | {--allow | --deny | --unset}`

For each read, write, execute, or delete permission, specify the delete, allow, deny, or unset option, or leave the current value as is.

Option	**Description**
delete	The user or group can delete the object.

	allow	The user or group can perform the specified action on the object.
	deny	The user cannot perform the specified action on the object. **Note:** You cannot deny permissions for a group.
	unset	The user's or group's permissions for this action on this object have not been specified. Note that if you set all of an object's permissions to unset and the user is not the ACL's owner, the user will be removed from the ACL.
`<permission-name>`	Specify read, write, execute, or delete permission for each user or group in the ACL for an object.	
	Permission	**Description**
	r	The user or group can read the object.
	w	The user or group can write to the object.
	x	The user or group can execute the object.
	d	The user or group can delete the object.
`--target=<target> | <target>`	The path to the object whose access control list permissions you are modifying.	

`[-r | --recursive]`	Recursive mode (optional). If you are configuring permissions for a directory, specify the -r option if you want the permission settings to apply recursively to all the directory's contents. If you do not specify the -r option, the settings apply only to the directory itself.
`--lock`	If you specify the --lock option, all objects in a directory are locked while the permissions on the directory are modified. Locking the objects reduces the time required to perform a chmod operation on a large share. Use the --lock option only if the chmod operation has exclusive access to the directory on which the permissions will be modified. If the chmod operation does not have exclusive access, there is a strong potential for a deadlock to occur. To provide exclusive access, stop all rehash, search, and other operations that involve the directory.

avaki chown

Change the owner of an object in the data catalog.

Syntax

```
avaki chown [-r | recursive] <user-grid-path>
  <target-grid-path>
```

`-r | recursive`	Recursive mode (optional). If you are changing ownership permissions for a directory, the system changes the permissions of all the directory's contents, including all subdirectories and their contents.
`<user-grid-path>`	The path of the user or group whose permissions you are changing in an object's access control list.
`<target-grid-path>`	The path to the file or directory whose ownership you are modifying.

avaki client

Connect an Avaki command-line client to an Avaki domain, disconnect a command client from a domain, or get information about a client connected to a domain.

Syntax

```
avaki client {--connect <machine>[:<port>] | --disconnect |
  --info}
```

`--connect <machine>[:<port>]`	Connect a command client to the grid domain controller or grid server running on <machine>. For <machine>, specify a DNS name or IP address. For <port>, specify the Avaki server's connect port number.
`--disconnect`	Disconnect a command client from an Avaki domain.
`--info`	Display the name of the Avaki domain to which a client is connected.

avaki cp

Copy files and directories (Avaki or local) from the specified source file or path to the specified destination. You can use **avaki cp** to copy from a grid directory to the local file system, from one grid directory to another, from the local file system to a shared directory, or from the local file system to the local file system. You may not copy from a shared directory to a nonshared directory. This command is similar to the Unix **cp** command.

Caution When you copy a file or directory into an Avaki share, out of an Avaki share, or within an Avaki share, the change is reflected in the source file system—not in the grid only. You cannot copy an entire shared directory.

Syntax

```
avaki cp [-r] [--localsrc] [--localdst] <copy-source>+
  <copy-destination>
```

`-r`	Recursive mode (optional). If the source or destination is a directory, the system copies the contents, including all subdirectories and their contents.
`[--localsrc]`	Optional. Indicates that the source file or path is in the local file system. If you omit this option, the system assumes that the source file or path is in a grid directory.
`[--localdst]`	Indicates that the destination file or path is in the local file system. If you omit this option, the system assumes that the destination file or path is in a grid directory.
`<copy-source>+`	One or more files, directories, or other objects to be copied. If you enter only a name, the system assumes that the object is in the current directory. Enter an absolute or relative path to copy an object in another directory.
`<copy-destination>`	The location for the copy. If you specify only a name, the system places the copy in the current directory. Specify an absolute or relative path to place the copy in another directory.

Example

This example copies the file bettysform.txt from the grid directory /Shares/Slaterock/ to the directory c:/Newforms on the local file system.

```
$ avaki cp --localdst /Shares/Slaterock/bettysform.txt
c:/Newforms/bettysform.txt
```

This example copies the file bettysform.txt from the grid directory /Shares/Slaterock to the grid directory /Shares/Newforms.

```
$ avaki cp /Shares/Slaterock/bettysform.txt
/Shares/Newforms/bettysform.txt
```

This example copies the file bettysform.txt from the local file system directory c:/Slaterock to the local file system directory c:/Newforms.

```
$ avaki cp --localsrc --localdst c:/Slaterock/bettysform.txt
c:/Newforms/bettysform.txt
```

avaki dataservice --add-schedule

Add a caching schedule for a data service.

To use this command, you must have write permission on the cache service, and you must have read and execute permission on the data service.

Syntax

```
avaki dataservice --add-schedule
  {--periodic-schedule=<period-spec> |
  --cron-schedule=<cron-spec> |
  --one-time-schedule=<time-stamp>}
  [--start-time=<time-stamp>] [[--end-time=<time-stamp>] |
  [--max-iterations=<num-iterations>]]
  [--time-zone=<time-zone>] [--exclusion=<exclusion-name>]+
  [--exclusion-action={ drop | sooner | later }]
  [--reschedule-offset=<offset-spec>] <cache-service-path>
  <qualified-ds-name> [<param-name>:<param-value>]+
```

`--periodic-schedule=<period-spec>`

Specify a periodic schedule for running the data service (which is identified in the <qualified-ds-name> argument). Use the following syntax:

```
--periodic-schedule="<number-of-units> <unit-type>"
```

`<number-of-units>`	An integer from 1 to 2147483648
`<unit-type>`	seconds, minutes, hours, days, weeks, or months

In the following example, the periodic schedule is set to 30 seconds:

```
--periodic-schedule="30 seconds"
```

`--cron-schedule=<cron-spec>`

Specify how often an advanced schedule based on a cron expression will run the data service (which is identified in the <qualified-ds-name> argument). Use the following syntax:

```
--cron-schedule="<seconds> <minutes> <hours> <days-of-month> <months> <days-of-week> [<years>]"
```

For details on cron expression syntax, see Appendix C, “Configuring cron schedules”.

`--one-time-schedule=<time-stamp>`

Specify a single time and date for running the data service (which is identified in the <qualified-ds-name> argument). Use the following syntax:

```
--one-time-schedule="{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> | now}"
```

`--start-time=<time-stamp>`

Specify the time and date that the data service schedule will start running, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> | now}
```

`--end-time=<time-stamp>`

Specify the time and date that the data service schedule will stop running, in this format:

`{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> | now}`

`--max-iterations=<num-iterations>`

Specify the number of times a schedule will repeat.

`<num-iterations>`	An integer from 1 to 2147483648. If no value is specified, the schedule repeats indefinitely.

`--time-zone=<time-zone>`

Specify the time zone for the schedule. The schedule can be specified according to the cache server's time zone or relative to Greenwich Mean Time (GMT). If no value is specified for the time zone, the server's local time zone is used. The format for the time zone is as follows:

GMT+12:00 Fiji	GMT+02:00 Cairo
GMT+11:00 Noumea	GMT+01:00 Paris
GMT+10:00 Sydney	GMT London
GMT+9:30 Adelaide	GMT-01:00 Azores
GMT+09:00 Tokyo	GMT-02:00 Mid-Atlantic
GMT+08:00 Hong Kong	GMT-03:00 Rio de Janeiro
GMT+07:00 Bangkok	GMT-03:30 Newfoundland
GMT+06:30 Rangoon	GMT-04:00 Caracas
GMT+06:00 Dacca	GMT-05:00 New York
GMT+05:45 Katmandu	GMT-06:00 Chicago
GMT+05:30 Calcutta	GMT-07:00 Denver
GMT+05:00 Karachi	GMT-08:00 San Francisco
GMT+04:30 Kabul	GMT-09:00 Juneau
GMT+04:00 Baku	GMT-10:00 Hawaii
GMT+03:30 Tehran	GMT-11:00 Samoa
GMT+03:00 Moscow	

`--exclusion=<exclusion-name>`

The name of an exclusion to this schedule. For information about creating schedule exclusions, see **avaki scheduleexclusion --create --daily** (page 201), **avaki scheduleexclusion --create --weekly** (page 205), **avaki scheduleexclusion --create --monthly** (page 203), **avaki scheduleexclusion --create --yearly** (page 207), and **avaki scheduleexclusion --create --custom** (page 199).

`--exclusion-action={drop | sooner | later}`

Specify the action to take if the data service is scheduled to run during an exclusionary period.

drop	Don't run the data service during the exclusionary period.
sooner	Reschedule the data service to run before the exclusionary period
later	Reschedule the data service to run after the exclusionary period.

`--reschedule-offset=<offset-spec>`

Specify the amount of time before or after the exclusionary period to perform a rescheduled data service run. Use the following syntax:

```
--reschedule-offset="<number-of-units>
<unit-type>"
```

`<number-of-units>`	An integer from 1 to 2147483648
`<unit-type>`	Seconds, minutes, hours, days, weeks, or months

`<cache-service-path>`

The data catalog path of a cache service.

`<qualified-ds-name>`

The name of a data service, in this format:

```
[<domain-name>.]<data-service-name>
```

`<param-name>:<param-value>+`

Optional. One or more parameters required by the data service and one or more parameter values.

Example

This example adds a periodic schedule in which the data service FlintstoneDataService is run every 30 seconds and the time zone is GMT-05:00 New York.

```
$ avaki dataservice --add-schedule --periodic-schedule=
"30 seconds" --time-zone="GMT-05:00 New York"
/System/LocalDomain/Services/CacheServices/
BEDROCK.sybase.com FlintstoneDataService
```

avaki dataservice --create

Create a data service.

To use this command, you must be a member of the DataProviders group.

Syntax

```
avaki dataservice --create [{--server | -s}=<grid-server>]
  [--localsrc] [--expiration=<expiration-time-in-secs>]
  [--run-user={<qualified-user-name> | null}]
  <descriptor-path> <data-service-name>
```

`--server | -s}=<grid-server>`	Optional. Specify the name of the Avaki server on which the data service will be created. For example, Bedrock. If you omit this option, the data service will be created on the same server as the parent directory.
`--localsrc`	Optional. Specify the --localsrc option if the <descriptor-path> is a path in the local file system. If --localsrc is not specified, <descriptor-path> is assumed to be a grid path.
`--expiration= <expiration-time-in-secs>`	Optional. Specify the interval, in seconds, after which the results of this data service expire from the on-demand cache. If you set this option to –1, the results never expire. If you set it to 0, the data is not cached. Default: 0 (results are not cached) Range: 0 to 2147483647

`--run-user={<qualified-user-name> | null}`

Specify the qualified name of the user to create the data service as. The format is:

```
<user>@<authservice>.<authservicetype>.<domain>
```

For example:

```
wilma@DefaultAuthService.Grid.Bedrock
```

The elements of <qualified-user-name> are as follows:

<user>	The name of the user to create the data service as.
<authservice>	The name of the user's authentication service.
<authservicetype>	The type of authentication service (Grid, Nis, or Ldap).
<domain>	The name of the user's Avaki domain.

If you specify null, the data service is run as the current user.

`<descriptor-path>`

Optional. Specify the path to an XML file that contains a data service descriptor. See the *Sybase Avaki EII Provisioning and Advanced Data Integration Guide* for a description of the XML schema that specifies the contents of a data service.

`<data-service-name>`

A name for the data service. The name must not contain any period characters (.) or spaces.

avaki dataservice --delete

Delete a data service.

To use this command, you must have execute permission on the data service.

Syntax

```
avaki dataservice --delete <qualified-ds-name>
```

`<qualified-ds-name>`	The name of a data service, in this format: `[<domain-name>.]<data-service-name>`

avaki dataservice --delete-schedule

Delete a caching schedule for a data service.

To use this command, you must have write permission on the cache service.

Syntax

```
avaki dataservice --delete-schedule <cache-service-path>
  <qualified-ds-name> [<param-name>:<param-value>]+
  <schedule-id>
```

`<cache-service-path>`	The data catalog path of a cache service.
`<qualified-ds-name>`	The name of a data service, in this format: `[<domain-name>.]<data-service-name>`
`<param-name>:<param-value>+`	Optional. One or more parameters required by the data service and one or more parameter values.
`<schedule-id>`	The numeric ID of the schedule to delete. To determine a schedule's ID, use **avaki dataservice --list-schedules** (page 58).

avaki dataservice --depends

Show the dependencies to and from a data service.

To use this command, you must have read permission on the message service on the grid server where the data service is located, and you must have read permissions on the data service.

Syntax

```
avaki dataservice --depends <qualified-ds-name>
```

`<qualified-ds-name>`	The name of a data service, in this format: `[<domain-name>.]<data-service-name>`

avaki dataservice --execute

Run a data service.

To use this command, you must have execute permission on the data service.

Syntax

```
avaki dataservice --execute <qualified-ds-name>
   [<param-name>:<param-value>]+ [--generate-schema]
```

`<qualified-ds-name>`	The name of a data service, in this format: `[<domain-name>.]<data-service-name>`
`<param-name>:<param-value>+`	Optional. Specify one or more input or output parameters and associated values required by the data service.
`--generate-schema`	Optional. Generate an XSD schema file for the data service. The schema file describes the structure of the output of the data service. The generated file will be stored in the following path and file: `/Metadata/DataService/<data-service-name>/<data-service-name>.xsd`

avaki dataservice --generate-sql-view

Generate a SQL view from a data service. The SQL view is placed in the following data catalog directory:

System/Domains/<domain-name>/Services/DataServices/VirtualSQLViews

Note You must generate a schema before generating a SQL view. You can use **avaki dataservice --execute** (page 56) to generate a schema.

Syntax

```
avaki dataservice --generate-sql-view
  [--provision-as=<table-name>]
  [--param-column=<input-param-name>:<column-name>]+
  <qualified-ds-name>
```

`--provision-as=<table-name>`	Optional. Specify a name for the SQL view. If no name is specified, the SQL view will have the same name as the data service.
`--param-column=<input-param-name>:<column-name>`	Optional. Specify column names for the input parameters, if any. When Avaki generates a SQL view for a data service, it must create table column names for any input parameters to the data service (in addition to using the column names for the output data specified in the schema file) so that the input parameters can be specified in a SQL query on that SQL view. If you omit this option, Avaki uses the name(s) of the data service input parameter(s) to create column names, first removing illegal characters such as spaces.
`<qualified-ds-name>`	The name of the data service for which you are generating a SQL view, in this format: `[<domain-name>.]<data-service-name>`

avaki dataservice --info

Display the XML data service descriptor.

Syntax

```
avaki dataservice --info <qualified-ds-name>
```

`<qualified-ds-name>`	The name of a data service, in this format: `[<domain-name>.]<data-service-name>`

avaki dataservice --list-schedules

List the caching schedules for a data service.

Syntax

```
avaki dataservice --list-schedules <cache-service-path>
  <qualified-ds-name> [<param-name>:<param-value>]+
```

`<cache-service-path>`	The grid path of a cache service.
`<qualified-ds-name>`	The name of a data service, in this format: `[<domain-name>.]<data-service-name>`
`<param-name>:<param-value>+`	Optional. Specify one or more input or output parameters and associated values required by the data service.

avaki dataservice --update

Modify the configuration of a data service.

Syntax

```
avaki dataservice --update <qualified-ds-name> [[--localsrc]
  <descriptor-path>] [--expiration=<expiration-time-in-secs>]
  [--run-user={<qualified-user-name> | null}]
```

`<qualified-ds-name>`	The name of a data service, in this format: `[<domain-name>.]<data-service-name>`
`--localsrc`	Optional. Use the --localsrc option if the <descriptor-path> is a path in the local file system. If --localsrc is not specified, <descriptor-path> is assumed to be a grid path.
`<descriptor-path>`	Optional. Specify the path to an XML file that contains a data service descriptor. See the *Sybase Avaki EII Provisioning and Advanced Data Integration Guide* for a description of the XML schema that specifies the contents of a data service.
`--expiration=<expiration-time-in-secs>`	Optional. Specify the interval, in seconds, after which the results of this data service expire from the on-demand cache. If you set this option to –1, the results never expire. If you set it to 0, the data is not cached. Default: 0 (results are not cached) Range: 0 to 2147483647

`--run-user={<qualified-user-name> | null}`

Specify the qualified name of the user to modify the data service as. The format is:

```
<user>@<authservice>.<authservice-type>.<domain>
```

For example:

```
wilma@DefaultAuthService.Grid.Bedrock
```

The elements of <qualified-user-name> are as follows:

<user>	The name of the user to modify the data service as.
<authservice>	The name of the user's authentication service.
<authservice-type>	The type of authentication service (Grid, Nis, or Ldap).
<domain>	The name of the user's Avaki domain.

If you specify null, the database operation is run as the current user.

avaki dbconn --allow-dbop-creation

Allow a user to create database operations that use a designated database connection.

Syntax

```
avaki dbconn --allow-dbop-creation <dbconn-name>
  {<qualified-user-name>}
```

`<dbconn-name>` — The name of the database connector.

`<qualified-user-name>` — Specify the qualified name of the user who can create a database operation that uses the database connection. The format is:

`<user>@<authservice>.<authservicetype>.<domain>`

For example:

`wilma@DefaultAuthService.Grid.Bedrock`

The elements of <qualified-user-name> are as follows:

<user>	The name of the user who can create a database operation that uses the database connection.
<authservice>	The name of the user's authentication service.
<authservice-type>	The type of authentication service (Grid, Nis, or Ldap).
<domain>	The name of the user's Avaki domain.

avaki dbconn --delete

Delete an Avaki database connector.

To use this command, you must be a member of the DatabaseAdministrators group.

Syntax

```
avaki dbconn --delete <dbconn-name>
```

`<dbconn-name>` The name of the database connector to be deleted.

avaki dbconn --disallow-dbop-creation

Prevent a user from creating database operations that use a designated database connector.

Note This command is intended to reverse the effect of **avaki dbconn --allow-dbop-creation** (page 61); it might not reverse the effects of permissions that have been set manually.

Syntax

```
avaki dbconn --disallow-dbop-creation <dbconn-name>
  {<qualified-user-name>}
```

`<dbconn-name>` The name of the database connector.

`<qualified-user-name>` Specify the qualified name of the user to prevent from creating database operations that uses the database connection. The format is:

`<user>@<authservice>.<authservicetype>.<domain>`

For example:

`wilma@DefaultAuthService.Grid.Bedrock`

The elements of <qualified-user-name> are as follows:

<user>	The name of the user to prevent from creating a database operation that use the database connection.
<authservice>	The name of the user's authentication service.
<authservice-type>	The type of authentication service (Grid, Nis, or Ldap).
<domain-name>	The user's grid domain.

avaki dbconn --execute

Execute an ad-hoc SQL statement on a database connector.

To use this command, you must be a member of the DatabaseAdministrators group.

Syntax

```
avaki dbconn --execute [{--xmlinputfile | --localxmlinputfile}]
  [--prepared | --callable] [--batch][--params={in | out |
  inout}:<parameter-types>] <dbconn-name> <sql>
  [<parameter-value>+]
```

`--xmlinputfile`	Optional. If the query requires an input file, use this option to specify the path to the file. For details about the format for an XML input file, see the *Sybase Avaki EII Provisioning and Advanced Data Integration Guide*
`--localxmlinput-file`	Optional. If the query requires an input file, use this option to specify the local path to the file. For details about the format for an XML input file, see the *Sybase Avaki EII Provisioning and Advanced Data Integration Guide*
`--prepared`	Optional. Specify this option to use a JDBC prepared statement to execute the SQL.
`--callable`	Optional. Specify this option if the SQL statement <sql> invokes a stored procedure in the underlying database.
`--batch`	Optional. Specify this option if the SQL statement <sql> is an update that can be used with JDBC batch mode.

`--params={in | out | inout}: <parameter-types>`

Optional. If the query accepts input parameters or produces output parameters, use this option to specify the data type for each input and output. There is no default.

Use in for input parameters, out for output parameters, and inout for parameters that can be both. If you specify multiple parameters, use semicolons (;) to separate them.

You can use the following <parameter-types>:

BIGINT	LONGVARBINARY
BINARY	LONGVARCHAR
BIT	NUMERIC
BOOLEAN	ORACLE_CURSOR
BLOB	OTHER
CHAR	REAL
CLOB	SMALLINT
DATE	TIME
DECIMAL	TIMESTAMP
DOUBLE	TINYINT
FLOAT	VARBINARY
INTEGER	VARCHAR

`<dbconn-name>`

The name of a database connector.

`<sql>`

Enter the SQL statement that the database connector will execute. Enclose the statement in quotation marks.

`<parameter-value>+`

Optional. One or more parameter values required by the target database connector. If a database query is associated with the database connector, the parameter values will be processed in the order specified in the database operation's query. Any parameter value that includes a space must be enclosed in quotation marks. Any string at the end of the command that is enclosed in quotation marks will be interpreted as a single value. (That is, enclose each parameter value in its own pair of quotation marks.)

avaki dbconn --info

Display information about an Avaki database connector, including its JDBC driver, connection string, user name, connection timeout, and whether it is using Avaki connection pooling.

To use this command, you must be a member of the DatabaseAdministrators group.

Syntax

```
avaki dbconn --info <dbconn-name>
```

`<dbconn-name>` The name of a database connector.

avaki dbconn --jdbc

Create or modify an Avaki database connector. A database connector allows database operations and their views to run queries on a relational database. In addition, a database connector can be used to provision SQL views.

To use this command, you must be a member of the DatabaseAdministrators group.

Syntax

```
avaki dbconn --jdbc {--create | --update}
  --driver=<jdbc-driver-class> --connUrl=<jdbc-connection-url>
  [--user=<jdbc-user-name>] [--password=<jdbc-password>]
  [--pool={true | false}] [--adhoc={true | false}]
  [--properties=<user-jdbc-properties>]
  [{--server | -s}=<grid-server>]
  [--description=<dbconn-description-text>]
  [--mapping-file=<grid-path>]
  [--database-name=<database-name>] [--admin-name=<DBA-name>]
  [--admin-org=<DBA-org>] [--admin-email=<DBA-email>]
  [--admin-phone=<DBA-phone>]
  [--xa-datasource-class=<class-name>
  [--xa-property="<name=value>"]+] <dbconn-name>
```

`--create`	Set up a new database connector.
`--update`	Modify an existing database connector.
`--driver=<jdbc-driver-class>`	Specify the JDBC driver that this database connector will use.
`--connUrl=<jdbc-connection-url>`	Specify the connect string for the JDBC driver that connects to your database.
`--user=<jdbc-user-name>`	Optional. Specify the name of a user account in the target database.
`--password=<jdbc-password>`	Optional. Specify the password for <jdbc-user-name>. If you're executing this command on a Unix system and the password contains non-alphanumeric characters (! or &, for instance), enclose the password in single quotation marks to prevent the shell from interpreting the special characters.
`--pool={true | false}`	Optional. Indicate whether this database connector will use (true) or not use (false) Avaki connection pooling. When connection pooling is enabled, database connections can be reused, which typically improves performance for JDBC applications. Set this option to true if your JDBC driver does *not* have built-in connection pooling. Default: false (no pooling)
`--adhoc={true | false}`	Optional. Set this option to true to enable ad hoc queries, which let you directly query a database in SQL. Otherwise, set to false. Default: false

`--user-property="<name=value>"`	Optional. Enter any driver- or database-specific properties that are needed to connect to the target database. For example, the command line syntax might look like this:
	`--user-property="User=fred"`
	`--user-property="Password=mypassword"`
	`--user-property="ArbitraryStringPropery=hello" --user-property="ArbitraryIntProperty=1"`

`--server | -s}= <grid-server>`	Optional. Specify the name of the database connector's grid server. For example, Bedrock. If you omit this option, the database connector is created or modified on the current grid server.
`--description= <dbconn-description -text>`	Optional. Enter details about this database connector; enclose the entire text string in quotation marks. This description is shown to users who display information about the database connector (see **avaki dbconn --info** on page 66). Consider including an explanation of what this database connector is for or where to get more information. Length limit: 255 characters
`--mapping-file= <grid-path>`	Optional. Provide the name and path to an XML file in the data catalog in which you specify mappings between SQL data types and the data types used by the Avaki query engine. The mappings will be used for this database connector only. For information on the format and content of the mapping file, see Appendix D, "Data type mapping files".
`--database-name= <database-name>`	Optional. Enter the name of the database that this database connector links to; enclose the entire text string in quotation marks. This text is shown to users who display information about the database connector (see **avaki dbconn --info** on page 66). Length limit: 255 characters
`--admin-name= <DBA-name>`	Optional. Enter the name of the database administrator for this dbconn's database; enclose the entire text string in quotation marks. This text is shown to users who display information about the database connector (see **avaki dbconn --info** on page 66). Length limit: 255 characters
`--admin-org= <DBA-org>`	Optional. Enter the organization to which the database administrator belongs; enclose the entire text string in quotation marks. This text is shown to users who display information about the database connector (see **avaki dbconn --info** on page 66). Length limit: 255 characters

`--admin-email=<DBA-email>`	Optional. Enter the e-mail address of the database administrator for this dbconn's database; enclose the text in quotation marks. This text is shown to users who display information about the database connector (see **avaki dbconn --info** on page 66). Length limit: 255 characters
`--admin-phone=<DBA-phone>`	Optional. Enter the database administrator's telephone number. This text is shown to users who display information about the database connector (see **avaki dbconn --info** on page 66). Length limit: 255 characters
`--xa-datasource-class=<class-name>`	Optional. If you plan to use this database connector for distributed transactions, enter the fully qualified class name of the XADataSource implementation provided by the database vendor. This class must reside in the JAR file for the database vendor's JDBC driver. For details on configuring your XA-compliant JDBC driver, see the documentation for the driver.

`--xa-property="<name=value>"`

Enter a string (enclosed in quotation marks) that contains the name and value of a connection property to be set on the XADataSource. This string should be in the form "<property-name>=<property-value>." The property names must correspond to the name of a setter method to be called on the XADataSource, with the "set" prefixes excluded. If the value of the property is an integer, it may be specified as a string and will be converted to an integer before the setter method is called. To specify multiple properties, repeat the --xa-property option.

For example, if the following calls should be made on the XADataSource:

```
source.setUser("fred");
source.setPassword("mypassword");
source.setArbitraryStringProperty("hello");
source.setArbitraryIntProperty(1);
```

Then the command line syntax might look like this:

```
--xa-property="User=fred"
--xa-property="Password=mypassword"
--xa-property="ArbitraryStringPropery=hello"
--xa-property="ArbitraryIntProperty=1"
```

See also the Examples section, below. For details on the particular XA connection properties to use with your XA-compliant JDBC driver, refer to the documentation for the driver.

`<dbconn-name>`

A name for the database connector. The name must not exceed 64 characters in length and must not contain any period characters (.) or spaces.

Examples

This command creates an XA-enabled database connector called myXAconn for a Sybase ASE database called dino. (XA-enabled database connectors can support distributed transactions.)

```
$ avaki dbconn --jdbc --create
--driver=com.sybase.jdbc3.jdbc.SybDriver
--connUrl=jdbc:sybase:tds:myDBserver:15000/dino
--user=barney --password=hoppy
--xa-datasource-class=com.sybase.jdbc3.jdbc.SybXADataSource
--xa-property="ServerName=myDBserver"
--xa-property="PortNumber=5000"
```

```
--xa-property="ResourceManagerType=2"
--xa-property="ResourceManagerName=connection"
--xa-property="DatabaseName=dino"
--xa-property="NetworkProtocol=Tds" --xa-property="User=barney"
--xa-property="Password=hoppy" myXAconn
```

This command creates an XA-enabled database connector called conn2pebbles for a MySQL database called Pebbles whose listening port is 17089.

```
$ avaki dbconn --jdbc --create
--driver=com.mysql.jdbc.Driver
--connUrl=jdbc:mysql://dbserver:17089/Pebbles
--user=barney --password=hoppy
--xa-datasource-class=com.mysql.jdbc.jdbc2.optional.MysqlXA
DataSource
--xa-property="URL=jdbc:mysql://dbserver/Pebbles"
--xa-property="User=barney" --xa-property="Password=hoppy"
conn2pebbles
```

This command creates a database connector called gazoo-conn for a DB2 database called gazoo whose listening port is 6789.

```
$ avaki dbconn --jdbc --create
--driver=COM.ibm.db2.jdbc.net.DB2Driver
--connUrl=jdbc:db2://localhost:6789/gazoo --user=barney
--password=hoppy gazoo-conn
```

avaki dbconn --provision-tables

Provision a SQL view from a database connector.

Syntax

```
avaki dbconn --provision-tables <dbconn-name>
  --spec=<provision-spec>+
```

`<dbconn-name>`	The name of the database connector from which you are provisioning a SQL view.
`--spec=<provision-spec>+`	Specify one or more parameters for provisioning the SQL view; enclose parameters in double quotation marks (for example, --spec="table=foo;type=TABLE"). The <provision-spec> options are as follows:

`provision-as=<provisioned-SQL-view-name>`	The name of the SQL view to provision.
`[;catalog=<database-catalog>]`	Optional. The name of the database catalog to provision.
`[;schema=<database-schema>]`	Optional. The name of the database schema to provision.
`[;table=<table-name>]`	Optional. The name of the table to provision.
`[;description=<description>]`	Optional. Enter details about the SQL view; enclose the entire text string in single quotation marks. Consider including an explanation of what this SQL view is for, how to use it, or where to get more information. Length limit: 255 characters
`[;type=<table-type>]*`	Optional. The type of table supported by the native database. The valid types are TABLE and VIEW.

avaki dbconn --test

Test an Avaki database connector. The test connects to the target database and displays the JDBC driver version, database version, and database name. To use this command, you must be a member of the DatabaseAdministrators group.

Syntax

```
avaki dbconn --test <dbconn-name>
```

`<dbconn-name>`	The name of the database connector to be tested.

avaki dbconn --show-tables

Display SQL views that have been provisioned from a database connector.

Syntax

```
avaki dbconn --show-tables <dbconn-name>
  [--catalog=<database-catalog>] [--schema=<database-schema>]
  [--table=<table-name>] [--type=<table-type>]*
```

`<dbconn-name>`	The name of the database connector whose provisioned SQL views you want to view.
`--catalog=<database-catalog>`	Optional. Restrict the display to SQL views from the specified catalog.
`--schema=<database-schema>`	Optional. Restrict the display to SQL views from the specified database schema.

`--table=<table-name>`	Optional. Restrict the display to SQL views from the specified table.
`--type=<table-type>`	Optional. Restrict the display to SQL views from the specified table type. The valid types are TABLE and VIEW.

avaki dbop --add-schedule

Add a caching schedule for a database operation.

To use this command, you must have write permission on the cache service, and you must have read and execute permission on the database operation.

Syntax

```
avaki dbop --add-schedule {--periodic-schedule=<period-spec> |
  --cron-schedule=<cron-spec> |
  --one-time-schedule=<time-stamp>}
  [--start-time=<time-stamp>] [[--end-time=<time-stamp>] |
  [--max-iterations=<num-iterations>]]
  [--time-zone=<time-zone>] [--exclusion=<exclusion-name>]+
  [--exclusion-action={ drop | sooner | later }]
  [--reschedule-offset=<offset-spec>] <cache-service-path>
  <qualified-dbop-name> [<parameter-value>]+
```

`--periodic-schedule=<period-spec>`	Specify a periodic schedule for running the database operation (which is identified in the <qualified-dbop-name> argument). Use the following syntax: `--periodic-schedule="<number-of-units> <unit-type>"` `<number-of-units>` — An integer from 1 to 2147483648 `<unit-type>` — seconds, minutes, hours, days, weeks, or months

In the following example, the periodic schedule is set to 30 seconds:

```
--periodic-schedule="30 seconds"
```

`--cron-schedule=<cron-spec>`

Specify how often a cron-based schedule for running the database operation (which is identified in the <qualified-dbop-name> argument) will run the database operation. Use the following syntax:

```
--cron-schedule="<seconds> <minutes> <hours>
<days-of-month> <months> <days-of-week>
[<years>]"
```

For details about the <cron-spec> syntax, see Appendix C, "Configuring cron schedules".

`--one-time-schedule=<time-stamp>`

Specify a single time and date for running the database operation (which is identified in the <qualified-dbop-name> argument). Use the following syntax:

```
--one-time-schedule="{<hh>:<mm>:<ss> {am | pm}
<mm>-<dd>-<yyyy> | now}"
```

`--start-time=<time-stamp>`

Specify the time and date that the database operation will start running, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> |
now}
```

`--end-time=<time-stamp>`

Specify the time and date that the database operation will stop running, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> |
now}
```

`--max-iterations=<num-iterations>`

Specify the number of times a schedule will repeat.

`<num-iterations>`	An integer from 1 to 2147483648. If no value is specified, the schedule repeats indefinitely.

`--time-zone=<time-zone>`

Specify the time zone for the schedule. The schedule can be specified according to the cache server's time zone or relative to Greenwich Mean Time (GMT). If no value is specified for the time zone, the server's local time zone is used. The format for the time zone is as follows:

GMT+12:00 Fiji	GMT+02:00 Cairo
GMT+11:00 Noumea	GMT+01:00 Paris
GMT+10:00 Sydney	GMT London
GMT+9:30 Adelaide	GMT-01:00 Azores
GMT+09:00 Tokyo	GMT-02:00 Mid-Atlantic
GMT+08:00 Hong Kong	GMT-03:00 Rio de Janeiro
GMT+07:00 Bangkok	GMT-03:30 Newfoundland
GMT+06:30 Rangoon	GMT-04:00 Caracas
GMT+06:00 Dacca	GMT-05:00 New York
GMT+05:45 Katmandu	GMT-06:00 Chicago
GMT+05:30 Calcutta	GMT-07:00 Denver
GMT+05:00 Karachi	GMT-08:00 San Francisco
GMT+04:30 Kabul	GMT-09:00 Juneau
GMT+04:00 Baku	GMT-10:00 Hawaii
GMT+03:30 Tehran	GMT-11:00 Samoa
GMT+03:00 Moscow	

`--exclusion=<exclusion-name>`

The name of an exclusion to this schedule. For information about creating schedule exclusions, see **avaki scheduleexclusion --create --daily** (page 201), **avaki scheduleexclusion --create --weekly** (page 205), **avaki scheduleexclusion --create --monthly** (page 203), **avaki scheduleexclusion --create --yearly** (page 207), and **avaki scheduleexclusion --create --custom** (page 199).

`--exclusion-action={drop | sooner | later}`

Specify the action to take if the database operation is scheduled to run during an exclusionary period.

drop	Don't run the database operation during the exclusionary period.
sooner	Reschedule the database operation to run before the exclusionary period.
later	Reschedule the database operation to run after the exclusionary period.

`--reschedule-offset=<offset-spec>`

Specify the amount of time before or after the exclusionary period to perform a rescheduled database operation run. Use the following syntax:

```
--reschedule-offset="<number-of-units> <unit-type>"
```

`<number-of-units>`	An integer from 1 to 2147483648
`<unit-type>`	Seconds, minutes, hours, days, weeks, or months

`<cache-service-path>`

The grid path of a cache service.

`<qualified-dbop-name>`

The name of a database operation, in this format:

```
[<domain-name>.]<database-connector-name>.<database-operation-name>
```

`<parameter-value>+`

Optional. One or more parameter values required by the target database operation. The parameter values will be processed in the order specified in the database operation's query. Any parameter value that includes a space must be enclosed in quotation marks. Any string at the end of the command that is enclosed in quotation marks will be interpreted as a single value. (That is, enclose each parameter value in its own pair of quotation marks.)

Example

This example adds a periodic schedule in which the database operation MyDBOP is run every 30 seconds and the time zone is GMT-05:00 New York.

```
$ avaki dbop --add-schedule --periodic-schedule="30 seconds"
--time-zone="GMT-05:00 New York" /System/LocalDomain/
Services/CacheServices/MAGADIEUXP.
sybase.com MyDBConnector.MyDBOP
```

avaki dbop --delete

Remove an Avaki database operation.

To use this command, you must be a member of the DatabaseAdministrators group.

Syntax

```
avaki dbop --delete <dbconn-name> <dbop-name>
```

<dbconn-name>	The name of the Avaki database connector that this database operation uses.
<dbop-name>	The name of the database operation to be deleted.

avaki dbop --delete-schedule

Delete a caching schedule for an Avaki database operation.

To use this command, you must have write permission on the cache service.

Syntax

```
dbop --delete-schedule <cache-service-path>
  <qualified-dbop-name> [<parameter-value>]+ <schedule-id>
```

`<cache-service-path>`	The grid path of a cache service.
`<qualified-dbop-name>`	The name of a database operation, in this format: `[<domain-name>.]<database-connector-name>.<database-operation-name>`
`<parameter-value>+`	Optional. One or more parameter values required by the target database operation. The parameter values must match the values that were specified when the schedule was created. The parameter values will be processed in the order specified in the database operation's query. Any parameter value that includes a space must be enclosed in quotation marks. Any string at the end of the command that is enclosed in quotation marks will be interpreted as a single value. (That is, enclose each parameter value in its own pair of quotation marks.)
`<schedule-id>`	The numeric ID of the schedule to delete. To determine a schedule's ID, use **avaki dbop --list-schedules** (page 91).

avaki dbop --depends

Show the dependencies to and from a database operation.

To use this command, you must have read permission on the message service on the grid server where the database operation is located, and you must have read permissions on the database operation.

Syntax

```
avaki dbop --depends {VirtualDb | <dbconn-name>} <dbop-name>
```

`VirtualDb | <dbconn-name>`	If you are showing the dependencies for a virtual SQL view that does a query against the query engine, enter the string VirtualDb. Otherwise, specify the name of the Avaki database connector for which you are showing the dependencies. The name of the Avaki virtual database or database connector that the database operation uses.
`<dbop-name>`	The name of the database operation for which dependencies will be shown.

avaki dbop --execute

Run an Avaki database operation.

To use this command, you must have execute permission on the database operation.

Syntax

```
avaki dbop --execute [{--xmlinputfile=<grid-file-path> |
  --localxmlinputfile=<local-file-path>}] [--generate-schema]
  {VirtualDb | [<dbconn-name>} <dbop-name> [<parameter-value>]+
```

`--xmlinputfile=<grid-file-path>`	Optional. If the database operation requires an input file, use this option to specify the grid path to the file. For details about the format for an XML input file, see the *Sybase Avaki EII Provisioning and Advanced Data Integration Guide*
`--localxmlinput-file=<local-file-path>`	Optional. If the database operation requires an input file, use this option to specify the local path to the file. For details about the format for an XML input file, see the *Sybase Avaki EII Provisioning and Advanced Data Integration Guide*
`--generate-schema`	Optional. Generate an XSD schema file for the database operation and suppress output. The schema file describes the structure of the output of a database operation. The generated file will be stored in the following path and file: `/Metadata/Database/<dbconn-name>/<dbop-name>/<dbopname>.xsd`
`VirtualDb | <dbconn-name>`	If you are running a virtual database operation that does a query against the query engine, enter the string VirtualDb.Otherwise, specify the name of the Avaki database connector for which you are running a database operation.

`<dbop-name>`	The name of the database operation to run.
`<parameter-value>+`	Optional. One or more parameter values required by the target database operation. The parameter values will be processed in the order specified in the database operation's query. Any parameter value that includes a space must be enclosed in quotation marks. Any string at the end of the command that is enclosed in quotation marks will be interpreted as a single value. (That is, enclose each parameter value in its own pair of quotation marks.)

Examples

This command runs a database operation called emp-info. It uses the database connector sybase1 and the parameter value "barney rubble".

```
$ avaki dbop --execute sybase1 emp-info "barney rubble"
```

This command runs a database operation called updatesals. It uses the database connector sybase1 and the input file c:\newsalaries.xml in the local file system.

```
$ avaki dbop --execute
--localxmlinputfile=c:\newsalaries.xml sybase1 updatesals
```

avaki dbop --generate-sql-view

Generate a SQL view from a database operation. The SQL view is placed in the following data catalog directory:

System/Domains/<domain-name>/Services/DatabaseServices/<database-connector-name>/VirtualSQLViews/

Note You must generate a schema before generating a SQL view. You can use **avaki dbop --execute** (page 81) to generate a schema.

Syntax

```
avaki dbop --generate-sql-view [--provision-as=<table-name>]
  {VirtualDb | <dbconn-name>} <dbop-name>
```

`--provision-as=<table-name>`	Optional. Specify a name for the SQL view. If no name is specified, the SQL view will have the same name as <dbop-name>.
`VirtualDb | <dbconn-name>`	If you create a virtual SQL view that does a query against the query engine, enter the string VirtualDb. Otherwise, specify the name of the Avaki database connector for which you are generating a SQL view.
`<dbop-name>`	The name of the database operation for which you are generating a SQL view.

avaki dbop --info

Display information about a database operation, including the name, description, SQL statement, parameters, and other options set when the database operation was created.

To use this command, you must be a member of the DatabaseAdministrators group.

Syntax

```
avaki dbop --info <dbconn-name> <dbop-name>
```

`<dbconn-name>`	The name of the Avaki database connector that the target database operation uses.
`<dbop-name>`	The name of a database operation.

avaki dbop --jdbc

Create or modify an Avaki database operation. (A database connector for this database operation must previously have been created, using either the web UI or the command **avaki dbconn --jdbc --create** (page 66)).

To use this command, you must be a member of the DatabaseAdministrators group.

Syntax

```
avaki dbop --jdbc {--create | --update}
  [--callable={true | false}] [--batch={true | false}]
  [--expiration=<expiration-time-in-secs>]
  [--params={in | out | inout}:<parameter-types>]
  [--description=<description-text>]
  [--run-user={<user-grid-path> | <qualified-user-name> |
  null}] [--sqlinputfile] <sql> <dbconn-name> <dbop-name>
```

`--create`	Create a new database operation.
`--update`	Modify an existing database operation.
`--callable={true | false}`	Optional. Set this option to true if the SQL statement <sql> invokes a stored procedure in the underlying database. Otherwise, set to false. Default: false
`--batch={true | false}`	Optional. Set this option to true if the SQL statement <sql> is an update that can be used with JDBC batch mode. Otherwise, set to false. Default: false
`--expiration=<expiration-time-in-secs>`	Optional. Specify the interval, in seconds, after which the results of this database operation expire from the on-demand cache. If you set this option to –1, the results never expire. If you set it to 0, the data is not cached. Default: 0 (results are not cached) Range: 0 to 2147483647

`--params={in | out | inout}:<parameter-types>`

Optional. If the database operation accepts input parameters or produces output parameters, use this option to specify the data type for each input and output. There is no default.

Use in for input parameters, out for output parameters, and inout for parameters that can be both. If you specify multiple parameters, use semicolons (;) to separate them.

You can use the following <parameter-types>:

BIGINT	LONGVARBINARY
BINARY	LONGVARCHAR
BIT	NUMERIC
BOOLEAN	ORACLE_CURSOR
BLOB	OTHER
CHAR	REAL
CLOB	SMALLINT
DATE	TIME
DECIMAL	TIMESTAMP
DOUBLE	TINYINT
FLOAT	VARBINARY
INTEGER	VARCHAR

`--description=<description-text>`

Optional. Enter details about this database operation; enclose the entire text string in quotation marks. The description is shown to users who display information about the database operation (see **avaki dbop --info** on page 83). Consider including an explanation of what this database operation is for, how to use it, or where to get more information.
Length limit: 255 characters

`--run-user={<user-grid-path> | <qualified-user-name> | null}`

Specify the user to create or modify the database operation as. You can specify the data catalog path to the directory of the user account. For example:

```
/System/Domains/Bedrock/Services/
AuthServices/Grid/
DefaultAuthService/Users/wilma
```

Alternatively, you can specify the qualified user name using the following format:

```
<user>@<authservice>.<authservicetype>.
<domain>
```

For example:

```
wilma@DefaultAuthService.Grid.Bedrock
```

The elements of <qualified-user-name> are as follows:

`<user>`	The name of the user to create or modify the database operation as.
`<authservice>`	The name of the user's authentication service.
`<authservicetype>`	The type of authentication service (Grid, Nis, or Ldap).
`<domain>`	The name of the user's Avaki domain.

If you specify null, the database operation is run as the current user.

`--sqlinputfile`

Optional. If --sqlinputfile is present, it indicates that the value specified in the <sql> argument is a path to a local file that contains a SQL statement that the database operation will execute. If --sqlinputfile is not present, <sql> consists of the SQL statement that the database operation will execute.

`<sql>`	Enter one of the following: • The SQL statement that the virtual database operation will execute. Enclose the statement in quotation marks, as shown in the examples below. • The path to a local file containing the SQL statement. If you choose this option, include --sqlinputfile in the command.
`<dbconn-name>`	The name of the Avaki database connector that will give this database operation access to a relational database.
`<dbop-name>`	The name of the database operation. The name must not exceed 64 characters in length and may not contain any period characters (.) or spaces.

Examples

This command creates a database operation called gazoobirthdays, which uses the database connector gazooconn to extract birth dates from an employee database.

```
$ avaki dbop --jdbc --create "select * from emp" gazooconn
gazoobirthdays
```

In the next example, the **avaki dbop --jdbc** command creates a database operation called employee, which uses the database connector sybase1 to extract employee records from a database. The --params option specifies that the database operation takes two input parameters, a varchar (the employee's name) and a numeric (the employee number).

The command **avaki dbop --execute** (page 81) runs the employee database operation, looking for the record of an employee named Barney whose employee number is 1234.

```
$ avaki dbop --jdbc --create --params=in:VARCHAR;in:NUMERIC
"select * from emp where ename=? AND empno=?" sybase1 employee

$ avaki dbop --execute sybase1 employee Barney 1234
```

avaki dbop --jdbc --create-virtual-dbop

Create an Avaki virtual database operation.

Syntax

```
avaki dbop --jdbc --create-virtual-dbop
  [--expiration=<expiration-time-in-secs>]
  [--params={in | out | inout}:<parameter-types>]
  [--description=<description-text>]
  [--run-user={<qualified-user-name> | null}] [--sqlinputfile]
  [--virtual-database=<virtual-database-path>]
  [{--server | -s}=<grid-server>] <sql> <dbop-name>
```

`--expiration=<expiration-time-in-secs>`

Optional. Specify the interval in seconds after which the results of this virtual database operation expire from the on-demand cache. If you set this option to –1, the results never expire. If you set it to 0, the data is not cached.
Default: 0 (results are not cached)
Range: 0 to 2147483647

`--params={in | out | inout}:<parameter-types>`

Optional. If the virtual database operation accepts input parameters or produces output parameters, use this option to specify the data type for each input and output. There is no default.

Use in for input parameters, out for output parameters, and inout for parameters that can be both. If you specify multiple parameters, use semicolons (;) to separate them.

You can use the following <parameter-types>:

BIGINT	LONGVARBINARY
BINARY	LONGVARCHAR
BIT	NUMERIC
BOOLEAN	ORACLE_CURSOR
BLOB	OTHER
CHAR	REAL
CLOB	SMALLINT
DATE	TIME
DECIMAL	TIMESTAMP
DOUBLE	TINYINT
FLOAT	VARBINARY
INTEGER	VARCHAR

`--description=<description-text>`

Optional. Enter details about this virtual database operation; enclose the entire text string in quotation marks. The description is shown to users who display information about the database operation (see **avaki dbop --info** on page 83). Consider including an explanation of what this virtual database operation is for, how to use it, or where to get more information. Length limit: 255 characters

`--run-user={<user-grid-path> | <qualified-user-name> | null}`

Specify the user to create or modify the virtual database operation as. You can specify the data catalog path to the directory of the user account. For example:

```
/System/Domains/Bedrock/Services/
AuthServices/Grid/
DefaultAuthService/Users/wilma
```

Alternatively, you can specify the qualified user name using the following format:

```
<user>@<authservice>.<authservicetype>.
<domain>
```

For example:

```
wilma@DefaultAuthService.Grid.Bedrock
```

The elements of <qualified-user-name> are as follows:

`<user>`	The name of the user to create or modify the virtual database operation as.
`<authservice>`	The name of the user's authentication service.
`<authservicetype>`	The type of authentication service (Grid, Nis, or Ldap).
`<domain>`	The name of the user's Avaki domain.

If you specify null, the virtual database operation is run as the current user.

`--sqlinputfile`	Optional. If --sqlinputfile is present, it indicates that the value specified in the <sql> argument is a path to a local file that contains a SQL statement that the database operation will execute. If --sqlinputfile is not present, <sql> consists of the SQL statement that the database operation will execute.
`--virtual-database=<virtual-database-path>`	Optional. The data catalog path to the virtual database service, in this format: `/System/LocalDomain/Services/VirtualDatabaseServices/Services/<grid-server>`
`{--server | -s}=<grid-server>`	Optional. Specify the name of the Avaki server on which the virtual database operation will be created. For example, Bedrock. If you omit this option, the virtual database operation is created on the current server.
`<sql>`	Enter one of the following: • The SQL statement that the virtual database operation will execute. Enclose the statement in quotation marks. • The path to a local file containing the SQL statement. If you choose this option, include --sqlinputfile in the command.
`<dbop-name>`	The name of the virtual database operation. The name must not exceed 64 characters in length and may not contain any period characters (.) or spaces.

avaki dbop --list-schedules

List the caching schedules for a database operation.

Syntax

```
avaki dbop --list-schedules <cache-service-path>
  <qualified-dbop-name> [<parameter-value>]+
```

`<cache-service-path>`	The data catalog path of a cache service.
`<qualified-dbop-name>`	The name of a database operation, in this format: `[<domain-name>.]<database-connector-name>.<database-operation-name>`
`<parameter-value>+`	Optional. One or more parameter values required by the target database operation. The parameter values will be processed in the order specified in the database operation's query. Any parameter value that includes a space must be enclosed in quotation marks. Any string at the end of the command that is enclosed in quotation marks will be interpreted as a single value. (That is, enclose each parameter value in its own pair of quotation marks.)

avaki dgas --add-admission-policy

Add a rule to the admission policy table for a data grid access server. To use this command, you must have execute permission on the target DGAS. (Use **avaki dgas --show-admission-policies** (page 114) to display the admission policies currently in effect. Use **avaki dgas --delete-admission-policy** (page 98) to remove rules from the policy table.)

An admission policy table is made up of an ordered list of rules that control which NFS and CIFS client machines may access the Avaki domain through this DGAS. The DGAS decides to permit or deny access by a client machine by matching the client's address with the policy rules. If the client address (either host or network) does not match the first rule, the DGAS looks at the next rule, continuing down the list until a match is found or the list ends. If a match is found (either admitting or denying the client), the DGAS acts accordingly. If no match is found in the rules but a default rule is defined, the client is admitted. If no match is found and no default is defined, the client is denied access. Note that rules are interpreted and applied in the order in which they appear in the table, so later rules can supersede earlier ones. However, deny rules always override allow rules.

Syntax

Two syntaxes perform the same actions:

```
dgas --add-admission-policy <dgas-name> <position> [--replace]
  {--deny --client-host=<machine> | --allow
  --client-host=<machine> --auth-service-path=<path>}

dgas --add-admission-policy <dgas-name> <position> [--replace]
  {--deny <client-machine> | --allow <client-machine>
  <auth-service-path>}
```

`<dgas-name>`	The name of a data grid access server.
`<position>`	The row number in the policy table at which you are adding the new rule. The rules are searched in row-number order. Row numbers start with 0.
`--replace`	Optional. When you use --replace, the new policy replaces the policy at the row number specified by <position>. If you don't use --replace, the new policy is added at the end of the last policy.

`--deny <client-machine>`
 	
or
 	
`--deny --client-host=<machine>`
	Deny access to the specified client machine or network. <client-machine> or <machine> are the IP address of the client or network in dotted decimal format (for example, 192.37.22.11 for a client or 192.37.22.0/24 for a network.).

`--allow <client-machine>`
 	
or
 	
`--allow --client-host=<machine>`
	Allow the specified client machine or network to access data through this DGAS. <client-machine> or <machine> are the IP address of the client or network in dotted decimal format (for example, 192.37.22.11 for a client or 192.37.22.0/24 for a network.).

`<auth-service-path>`
 	
or
 	
`--auth-service-path=<path>`
	Required when you use --allow. Specify the Avaki directory of the authentication service to be used for users and groups accessing this Avaki domain from the client(s) at the address specified in the --allow option. For example, /System/Domains/Bedrock/Services/AuthServices/Ldap/BedrockLdap.

avaki dgas --add-group-mapping

Map a group in your Avaki domain to a group ID (GID) on an NFS or CIFS client machine. This mapping is specific to the target DGAS and supersedes mappings configured in authentication services.

To use this command, you must have write permission on the target DGAS.

Syntax

```
avaki dgas --add-group-mapping <dgas-name> <gid> <client-host>
  <group-path>
```

`<dgas-name>`	The name of a data grid access server.
`<gid>`	The numeric group ID of the group on the client.
`<client-host>`	The IP address of the client machine in IPv4 dotted decimal format. (For example, 192.37.22.12.)
`<group-path>`	The data catalog path of the Avaki group.

avaki dgas --add-user-mapping

Map a user in your Avaki domain to a user ID (UID) on an NFS or CIFS client machine. This mapping is specific to the target DGAS and supersedes mappings configured in authentication services.

To use this command, you must have write permission on the target DGAS.

Syntax

```
avaki dgas --add-user-mapping <dgas-name> <uid> <client-host>
  <user-path>
```

`<dgas-name>`	The name of a data grid access server.
`<uid>`	The numeric user ID of the user on the client.
`<client-host>`	The IP address of the client machine in IPv4 dotted decimal format. (For example, 192.37.22.12.)
`<user-path>`	The data catalog path of the Avaki user.

avaki dgas --cifs-share-info

Display information about a CIFS share. To use this command, you must have read permission on the target CIFS share.

Syntax

```
avaki dgas --cifs-share-info <dgas-name> <share-name>
```

`<dgas-name>`	The name of a data grid access server.
`<share-name>`	The name of a CIFS share.

avaki dgas --clear-cached-credentials

Use this command when changes have occurred in authentication services, user accounts, group membership, user mappings, or group mappings.

To use this command, you need execute permission on the target DGAS.

Syntax

```
avaki dgas --clear-cached-credentials <dgas-name>
```

`<dgas-name>` The name of a data grid access server.

avaki dgas --create-cifs-share

Create a CIFS share to make data in the Avaki domain available to client machines.

To use this command, you must have execute permission on the target DGAS.

Caution Because the Windows file system on CIFS clients is case-insensitive, the <share-name> you assign to a CIFS share will be converted to all upper-case letters. Do not assign a share name that relies on case for uniqueness.

Syntax

```
avaki dgas --create-cifs-share <dgas-name> <share-name>
  <source-path> [--max-connections=<num>]
  [--enabled=<true | false>] [--comment=<string>]
```

`<dgas-name>`	The name of the data grid access server on which the CIFS share will reside.
`<share-name>`	A name for the CIFS share you are creating.
`<source-path>`	The path to the data catalog directory you are sharing out.
`--max-connections=<num>`	Optional. The maximum number of users that will be permitted to access this share at one time. To allow unlimited connections, set max connections to –1. Default: –1
`--enabled=<true | false>`	Optional. Turn this share on (true) or off (false). Default: true (the share is enabled)
`--comment=<string>`	Optional. Provide useful information such as a description of this CIFS share, an explanation of why this CIFS share exists, or contact information for the person responsible for it. This description will be visible to users on the CIFS client.

avaki dgas --delete-admission-policy

On the target data grid access server, remove the rule at the specified position in the admission policy table, or remove all the rules in the table.

To use this command, you must have execute permission on the target DGAS.

Use **avaki dgas --show-admission-policies** (page 114) to display the admission policy table.

Syntax

```
avaki dgas --delete-admission-policy <dgas-name> {<position> |
  --all}
```

`<dgas-name>`	The name of a data grid access server.
`<position>`	The row number of the rule to be removed.
`--all`	Remove all the rules in the admission policy table.

avaki dgas --delete-cache

Remove a specified file or directory from the internal cache of a data grid access server.

To use this command, you must have execute permission on the target DGAS.

Syntax

```
avaki dgas --delete-cache <dgas-name> [--target-path=<path>]
  [-r | --recursive]
```

`<dgas-name>`	The name of a data grid access server.
`--target-path=<path>`	Optional. Specify the data catalog path of the file or directory to be removed from the cache. If you omit this option, everything is removed from the cache.
`-r | --recursive`	Optional. If --target-path specifies a directory, recursively remove the directory and all its contents.

avaki dgas --delete-cifs-share

Remove a CIFS share.

To use this command, you must have delete permission on the target CIFS share.

Syntax

```
avaki dgas --delete-cifs-share <dgas-name> <share-name>
```

`<dgas-name>`	The name of a data grid access server.
`<share-name>`	The name of the CIFS share to be deleted.

avaki dgas --delete-group-mapping

Delete the mapping between the specified group ID on the specified client and the Avaki group it is mapped to. (That is, delete a mapping created by **avaki dgas --add-group-mapping** (page 94). This command does not affect mappings created in authentication services.) Use **avaki dgas --list-group-mappings** (page 106) to display a list of group mappings for a DGAS.

To use this command, you must have write permission on the target DGAS.

Syntax

```
avaki dgas --delete-group-mapping <dgas-name> [<client-machine>
   [<gid>]]
```

`<dgas-name>`	The name of a data grid access server.
`<gid>`	Optional if you use the <client-machine> option. Specify the group ID of the group on the client whose Avaki group mapping should be deleted. If you omit this option, the command deletes all group mappings for the specified client.
`<client-machine>`	The IP address of the client machine in IPv4 dotted decimal format. (For example, 192.37.22.12.)

avaki dgas --delete-user-mapping

Delete the mapping between the specified user ID on the specified client and the Avaki user it is mapped to. (That is, delete a mapping created by **avaki dgas --add-user-mapping** (page 95). This command does not affect mappings created in authentication services.) Use **avaki dgas --list-user-mappings** (page 107) to display a list of user mappings for a DGAS.

To use this command, you must have write permission on the target DGAS.

Syntax

```
avaki dgas --delete-user-mapping <dgas-name> [<client-machine>
  [<uid>]]
```

`<dgas-name>`	The name of a data grid access server.
`<uid>`	Optional if you use the <client-machine> option. Specify the user ID of the user on the client. If you omit this option, the command deletes all user mappings for the specified client.
`<client-machine>`	The IP address of the client machine in IPv4 dotted decimal format. (For example, 192.37.22.12.)

avaki dgas --disconnect-cifs-client

Disconnect a CIFS client from a data grid access server. If the client has more than one connection to the share, all connections are dropped.

To use this command, you must have execute permission on the target DGAS.

Syntax

```
avaki dgas --disconnect-cifs-client <dgas-name> <share-name>
  <client-host>
```

`<dgas-name>`	The name of the data grid access server on which the CIFS share was created.
`<share-name>`	The name of the CIFS share to be disconnected.
`<client-host>`	The DNS name or IP address of the client from which you are disconnecting the CIFS share.

avaki dgas --get-cache-size

Display the current size in bytes of the internal cache for the specified data grid access server.

To use this command, you must have read permission on the target DGAS.

Syntax

```
avaki dgas --get-cache-size <dgas-name>
```

`<dgas-name>` The name of a data grid access server.

avaki dgas --get-cache-statistics

Display statistics for the internal cache of the specified data grid access server. Statistics include cache hit and miss totals.

To use this command, you must have read permission on the target DGAS.

Syntax

```
avaki dgas --get-cache-statistics <dgas-name>
```

`<dgas-name>` The name of a data grid access server.

avaki dgas --get-free-disk-space

Display the amount of free disk space on the device on which the specified data grid access server's cache directory resides.

Syntax

```
avaki dgas --get-free-disk-space <dgas-name>
```

`<dgas-name>`	The name of a data grid access server.

avaki dgas --get-properties

For the specified data grid access server, list all DGAS properties and their values.

To use this command, you must have read permission on the target DGAS.

To change DGAS property values, use **avaki dgas --set-property** (page 114).

Syntax

```
avaki dgas --get-properties <dgas-name>
```

`<dgas-name>`	The name of a data grid access server.

avaki dgas --get-property

For the specified data grid access server, display the value of the specified property.

To use this command, you must have read permission on the target DGAS.

To change DGAS property values, use **avaki dgas --set-property** (page 114).

Syntax

```
avaki dgas --get-property <dgas-name> <prop-name>
```

`<dgas-name>`	The name of a data grid access server.
`<prop-name>`	The name of a DGAS property.

avaki dgas --get-property-list

Display descriptions of all DGAS properties, including default values and other information.

To use this command, you must have read permission on the target DGAS.

Syntax

```
avaki dgas --get-property-list <dgas-name>
```

`<dgas-name>`	The name of a data grid access server.

avaki dgas --initialize

Initialize credentials for a new data grid access server. (For an outline of the tasks involved in setting up a DGAS, see **avaki server --dgas --connect** on page 224.)

Syntax

```
avaki dgas --initialize <dgas-name>
  [--properties-file=<filename>]
```

`<dgas-name>`	A name for the new data grid access server.
`--properties-file=<filename>`	Optional. Specify the local path to the properties file the new DGAS will use. If you omit this option, the DGAS uses default values for all properties. Use **avaki dgas --get-property-list** (above) to display descriptions and default values of DGAS properties.

avaki dgas --list-cifs-clients

Display a list of addresses of client machines that are connected to the specified CIFS share.

To use this command, you must have read permission on the target DGAS.

Syntax

```
avaki dgas --list-cifs-clients <dgas-name> <share-name>
```

`<dgas-name>`	The name of the data grid access server on the which the CIFS share was created.
`<share-name>`	The name of the CIFS share whose connected clients you want to list.

avaki dgas --list-cifs-shares

Display a list of the CIFS shares that reside on the specified data grid access server.

To use this command, you must have read permission on the target DGAS.

Syntax

```
avaki dgas --list-cifs-shares <dgas-name>
```

`<dgas-name>` The name of a data grid access server.

avaki dgas --list-group-mappings

Display a list of the mappings between grid groups and client GIDs that are configured for the specified data grid access server. (Mappings configured in authentication services are not included.)

To use this command, you must have read permission on the target DGAS.

Syntax

```
avaki dgas --list-group-mappings <dgas-name>
```

`<dgas-name>` The name of a data grid access server.

avaki dgas --list-user-mappings

Display a list of the mappings between grid user accounts and client UIDs that are configured for the specified data grid access server. (Mappings configured in authentication services are not included.)

To use this command, you must have read permission on the target DGAS.

Syntax

```
avaki dgas --list-user-mappings <dgas-name>
```

`<dgas-name>`	The name of a data grid access server.

avaki dgas --modify-cifs-share

Change a CIFS share.

To use this command, you must have write permission on the target CIFS share.

Syntax

```
avaki dgas --modify-cifs-share <dgas-name> <share-name>
  [--name=<string>] [--max-connections=<num>]
  [--enabled=<true | false>] [--comment=<string>]
```

`<dgas-name>`	The name of the data grid access server on which the CIFS share resides.
`<share-name>`	The name of the CIFS share you are modifying.
`--name=<string>`	Optional. If you are changing the name of the CIFS share, use this option to specify a new name.

`--max-connections=<num>`	Optional. The maximum number of users that will be permitted to access this share at one time. A setting of –1 allows unlimited connections. Default: –1
`--enabled=<true | false>`	Optional. Turn this share on (true) or off (false). Default: true (the share is enabled)
`--comment=<string>`	Optional. Provide useful information such as a description of this CIFS share, an explanation of why this CIFS share exists, or contact information for the person responsible for it. This description will be visible to users on the CIFS client.

avaki dgas --read-log-properties

Force the target data grid access server to reread its log4j properties file, which is located in the local directory where the DGAS was started (that is, where the dgas --start command was executed).

To use this command, you must have execute permission on the target DGAS.

Syntax

```
avaki dgas --read-log-properties <dgas-name>
```

`<dgas-name>`	The name of a data grid access server.

avaki dgas --reset-cache-statistics

Reset all statistics counters for the internal cache of the specified data grid access server.

To use this command, you must have write permission on the target DGAS. (Use **avaki dgas --get-cache-statistics** (page 102) to display cache statistics.)

Syntax

```
avaki dgas --reset-cache-statistics <dgas-name>
```

`<dgas-name>`	The name of a data grid access server.

avaki dgas --save-cache

Save a copy of a data grid access server's internal cache to a specified local file.

To use this command, you must have execute permission on the target DGAS.

Syntax

```
avaki dgas --save-cache <dgas-name> <local-dest>
```

`<dgas-name>`	The name of a data grid access server.
`<local-dest>`	A directory name for the saved copy of the cache. The system creates the directory in the local cache directory of the target DGAS. The destination directory must not exist; the command creates it.

Example

Suppose the cache directory of the DGAS called Dinosdgas is /home/avaki/dgas/cache. If you enter this command, a copy of the cache is saved to /home/avaki/dgas/cache/2003-6-11:

```
$ avaki dgas --save-cache Dinosdgas 2003-6-11
```

avaki dgas --self-map

Creates a temporary dynamic mapping between the Avaki user who issues the command and the UID of the Unix account under which the issuing user is logged in. When a user is mapped to a UID, the reference count for that mapping is set to 1. The reference count is incremented each time a user is mapped to that UID. The reference count keeps track of how many jobs or processes are using the mapping. Unless you specify otherwise, dynamic mappings expire after 6 hours. Mappings created with this command are effective only on the DGAS specified by <dgas-name>.

User mappings allow local users to access data in the Avaki domain as mapped Avaki users. The dynamic mappings created with this command are especially useful when you want to use Avaki compute grid to run jobs that access data through the Avaki domain. Dynamic mappings are also useful in environments where IP addresses are assigned dynamically.

You can map the same Avaki user to several UIDs as long as the UIDs are on different machines. Two local accounts on the same machine cannot be mapped to the same Avaki user.

This command is available only on Unix systems.

Note You can't create a dynamic user mapping for a UID for which an Avaki administrator has already configured a DGAS-specific user mapping.

Note We recommend against using this command on a machine with more than one IP address. The mapping uses the machine's IP address to ensure that the UID is unique. On a multihomed machine, it is not possible to predict which IP address the mapping will use, which can lead to unexpected results.

Syntax

```
avaki dgas --self-map <dgas-name> [--time-to-live=<minutes>]
```

`<dgas-name>`	The name of a data grid access server.
`--time-to-live=<minutes>`	Optional. Specify the number of minutes this user-to-UID mapping will last. To set up a mapping that lasts indefinitely, set --time-to-live=–1. Default: 360 (6 hours)

avaki dgas --self-unmap

Removes (or decrements the reference count for) dynamic user mappings for the issuing user created with **avaki dgas --self-map** (above).

You can configure several compute grid jobs to use a single dynamic user mapping. To do this, execute **avaki dgas --self-map** once for each job. Then include an **avaki dgas --self-unmap** command at the end of each job. As the jobs finish running, each one decrements the reference count, until the last job removes the user mapping.

Syntax

```
avaki dgas --self-unmap <dgas-name> [--force]
```

`<dgas-name>`	The name of a data grid access server.
`--force`	Optional. Removes the mapping for the Avaki user who issues the command even if the reference count for the mapping is greater than 1.

avaki dgas --set-admission-policy-domain

Configure Windows domain information for the LDAP authentication service associated with an admission policy allow rule. (The target rule must not be a deny rule, because deny rules do not have associated authentication services.)

The DGAS supports pass-through authentication for LDAP authentication services. (That is, the DGAS forwards user names and passwords to a specified Windows domain controller for authentication.) If you do not set an admission policy domain, the DGAS uses the LDAP server for this grid domain's LDAP authentication service in place of the Windows domain controller.

To use this command, you must have execute permission on the target DGAS. To remove Windows domain information from an admission policy rule, use **avaki dgas --unset-admission-policy-domain** (page 117). To display Windows domain information for an admission policy rule, use **avaki dgas --show-admission-policy-domain** (page 115).

Syntax

```
avaki dgas --set-admission-policy-domain <dgas-name> <position>
  <domain-name> <domain-controller>+
```

`<dgas-name>`	The name of a data grid access server.
`<position>`	The row number in the admission policy table of the allow rule for which you're configuring domain information. (All allow rules are associated with grid authentication services.)
`<domain-name>`	A Windows domain name.
`<domain-controller>+`	The names of one or more Windows domain controllers for the specified Windows <domain-name>.

avaki dgas --set-default-admission-policy

Add or replace a default rule in the admission policy table for the specified data grid access server. The default rule is always the last one to be processed; it allows all clients who do not match any of the other rules to access the Avaki domain. (If you want access to be denied to all clients who are not specifically allowed by other rules, do not configure a default rule.) To use this command, you must have execute permission on the target DGAS. You can use **avaki dgas --delete-admission-policy** (page 98) to remove the default rule or any other rule from the policy table. Use **avaki dgas --show-admission-policies** (page 114) to display all the rules.

Syntax

```
avaki dgas --set-default-admission-policy <dgas-name>
  <auth-service-path>
```

`<dgas-name>`	The name of a data grid access server.
`<auth-service-path>`	The path to the grid authentication service that will provide user and group mappings for users of clients admitted by this rule.

avaki dgas --set-property

Set a property on a data grid access server.

To use this command, you must have write permission on the target DGAS.

Use **avaki dgas --get-property-list** (page 104) to display a list of DGAS properties and their values.

Syntax

```
avaki dgas --set-property <dgas-name> <prop-name> <value>
```

`<dgas-name>`	The name of a data grid access server.
`<prop-name>`	The name of a DGAS property.
`<value>`	The value you are assigning to <prop-name> for this DGAS.

avaki dgas --show-admission-policies

Display, in the configured order, the admission policies for the specified data grid access server. To use this command, you must have execute permission on the target DGAS.

Syntax

```
avaki dgas --show-admission-policies <dgas-name>
```

`<dgas-name>`	The name of a data grid access server.

avaki dgas --show-admission-policy-domain

Display Windows domain information for an authentication service in an admission policy rule for a data grid access server. The target rule must be a “deny” rule.

To use this command, you must have read permission on the target DGAS.

Syntax

```
avaki dgas --show-admission-policy-domain <dgas-name>
  <position>
```

`<dgas-name>`	The name of a data grid access server.
`<position>`	The row number in the admission policy table of the deny rule for which you’re displaying domain information.

avaki dgas --sync-cache

Check whether the DGAS's cached copy of the target file or directory is the most recent version. If the cached copy is not current or if no copy exists in the DGAS internal cache, the DGAS fetches the most recent version and caches it. (If a cache service for this DGAS has been configured, the DGAS tries to get the file or directory from the cache. If there is no cache or the cache does not have a current copy of the file or directory, the DGAS fetches it from the source file system.)

To use this command, you must have execute permission on the target DGAS and read permission on the target files and directories.

Syntax

```
avaki dgas --sync-cache <dgas-name> <target-path> [-r |
  --recursive]
```

`<dgas-name>`	The name of a data grid access server.
`<target-path>`	The grid path of a file or directory to be fetched into the cache.
`-r | --recursive`	Optional. If the target path is a directory, recursively fetch and cache all contents.

avaki dgas --unset-admission-policy-domain

Remove Windows domain information from an admission policy rule for a data grid access server. The target rule must be a "deny" rule.

To use this command, you must have execute permission on the target DGAS.

Syntax

```
avaki dgas --unset-admission-policy-domain <dgas-name>
  <position>
```

`<dgas-name>`	The name of a data grid access server.
`<position>`	The row number in the admission policy table of the deny rule from which you're removing domain information.

avaki dgas --unset-property

Unset the specified data grid access server property (that is, change the value to "unset"). When a DGAS property is unset, its value returns to the default setting.

Syntax

```
avaki dgas --unset-property <dgas-name> <prop-name>
```

`<dgas-name>`	The name of a data grid access server.
`<prop-name>`	The name of the DGAS property that you are unsetting.

avaki directory --add-schedule

Add a new pin or cache schedule to a currently cached Avaki directory or directory service. For information about tagging an Avaki directory or a directory hierarchy for caching within a cache service, see **avaki directory --cache** (page 122).

To use this command, you must have write permission on the cache service, and you must have read permission on the directory.

Syntax

```
avaki directory --add-schedule --action={directory-only |
  tree-on-demand | tree-pinned}
  {--periodic-schedule=<period-spec> |
  --cron-schedule=<cron-spec> |
  --one-time-schedule=<time-stamp>}
  [--start-time=<time-stamp>] [[--end-time=<time-stamp>] |
  [--max-iterations=<num-iterations>]]
  [--time-zone=<time-zone>] [--exclusion=<exclusion-name>]+
  [--exclusion-action={ drop | sooner | later }]
  [--reschedule-offset=<offset-spec>] <cache-service-path>
  <directory-path>+
```

`--action`	Specify how to cache the directory. The formats are as follows:	
	`<directory-only>`	Pin a directory but not its contents for scheduled caching.
	`<tree-on-demand>`	Tag a directory and its contents for on-demand caching, in which files or file portions can be cached as needed.
	`<tree-pinned>`	Pin a directory and its contents for scheduled caching.

`--periodic-schedule=<period-spec>`

Specify a periodic schedule for refreshing the cached directory data. Use the following syntax:

```
--periodic-schedule="<number-of-units> <unit-type>"
```

`<number-of-units>`	An integer from 1 to 2147483648
`<unit-type>`	seconds, minutes, hours, days, weeks, or months

In the following example, the periodic schedule is set to 30 seconds:

```
--periodic-schedule="30 seconds"
```

`--cron-schedule=<cron-spec>`

Create an advanced schedule using a cron expression that tells the cache service how often to refresh cached directory data. Use the following syntax:

```
--cron-schedule="<seconds> <minutes> <hours> <days-of-month> <months> <days-of-week> [<years>]"
```

For details about the <cron-spec> syntax, see Appendix C, “Configuring cron schedules”.

`--one-time-schedule=<time-stamp>`

Specify a single time and date for refreshing the cached directory data. Use the following syntax:

```
--one-time-schedule="{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> | now}"
```

`--start-time=<time-stamp>`

Specify the time and date that the cache service will start refreshing cached directory data, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> | now}
```

`--end-time=<time-stamp>`

Specify the time and date that the cache service will stop refreshing cached directory data, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> | now}
```

`--max-iterations=<num-iterations>` Specify the number of times a schedule will repeat.

`<num-iterations>`	An integer from 1 to 2147483648. If no value is specified, the schedule repeats indefinitely.

`--time-zone=<time-zone>` Specify the time zone for the schedule. The schedule can be specified according to the cache server's time zone or relative to Greenwich Mean Time (GMT). If no value is specified for the time zone, the server's local time zone is used. The format for the time zone is as follows:

GMT+12:00 Fiji	GMT+02:00 Cairo
GMT+11:00 Noumea	GMT+01:00 Paris
GMT+10:00 Sydney	GMT London
GMT+9:30 Adelaide	GMT-01:00 Azores
GMT+09:00 Tokyo	GMT-02:00 Mid-Atlantic
GMT+08:00 Hong Kong	GMT-03:00 Rio de Janeiro
GMT+07:00 Bangkok	GMT-03:30 Newfoundland
GMT+06:30 Rangoon	GMT-04:00 Caracas
GMT+06:00 Dacca	GMT-05:00 New York
GMT+05:45 Katmandu	GMT-06:00 Chicago
GMT+05:30 Calcutta	GMT-07:00 Denver
GMT+05:00 Karachi	GMT-08:00 San Francisco
GMT+04:30 Kabul	GMT-09:00 Juneau
GMT+04:00 Baku	GMT-10:00 Hawaii
GMT+03:30 Tehran	GMT-11:00 Samoa
GMT+03:00 Moscow	

`--exclusion=<exclusion-name>`

The name of an exclusion to this schedule. For information about creating schedule exclusions, see **avaki scheduleexclusion --create --daily** (page 201), **avaki scheduleexclusion --create --weekly** (page 205), **avaki scheduleexclusion --create --monthly** (page 203), **avaki scheduleexclusion --create --yearly** (page 207), and **avaki scheduleexclusion --create --custom** (page 199).

`--exclusion-action={drop | sooner | later}`

Specify the action to take if the cache service is scheduled to refresh cached directory data during an exclusionary period.

drop	Don't refresh the cached directory data during the exclusionary period.
sooner	Reschedule the cache service to refresh the cached directory data before the exclusionary period.
later	Reschedule the cache service to refresh the cached directory data after the exclusionary period.

`--reschedule-offset=<offset-spec>`

Specify the amount of time before or after the exclusionary period to perform a rescheduled cache refresh. Use the following syntax:

```
--reschedule-offset="<number-of-units> <unit-type>"
```

`<number-of-units>`	An integer from 1 to 2147483648
`<unit-type>`	Seconds, minutes, hours, days, weeks, or months

`<cache-service-path>`

The grid path of a cache service.

`<directory-path>+`

The grid path of one or more directories to which a pin schedule will be added.

This example pins the directory Water Buffalo Lodge and its contents for scheduled caching. The cached directory data is refreshed every 30 seconds and the time zone is GMT-05:00 New York.

```
$ avaki directory --add-schedule --action=tree-pinned
--periodic-schedule="30 seconds" --time-zone="GMT-05:00 New
York" /System/LocalDomain/Services/CacheServices/
MAGADIEUXP.sybase.com "/Shares/Water Buffalo Lodge"
```

avaki directory --cache

Tag an Avaki directory or a directory hierarchy for caching within a cache service.

Syntax

```
avaki directory --cache --action={directory-only |
  tree-on-demand | tree-pinned} [--offline-expiration=<value>]
  [{--periodic-schedule=<period-spec> |
  --cron-schedule=<cron-spec> |
  --one-time-schedule=<time-stamp>}]
  [--start-time=<time-stamp>] [[--end-time=<time-stamp>] |
  [--max-iterations=<num-iterations>]]
  [--time-zone=<time-zone>] [--exclusion=<exclusion-name>]+
  [--exclusion-action={ drop | sooner | later }]
  [--reschedule-offset=<offset-spec>] <cache-service-path>
  <directory-path>+
```

`--action`	Specify how to cache the directory. The formats are as follows:	
	`<directory-only>`	Pin a directory but not its contents for scheduled caching.
	`<tree-on-demand>`	Tag a directory and its contents for on-demand caching, in which files or file portions can be cached as needed.
	`<tree-pinned>`	Pin a directory and its contents for scheduled caching.

`--offline-expiration=<value>`

Optional. If the back-end source directory is offline, the cache service continues to return the cached data for the directory for the duration (in seconds) specified by the offline expiration value. A value of 0 indicates that the cache service is not allowed to return cached data if the source is offline. A value of -1 indicates that the cached data will never expire when the back-end source is offline. If the offline expiration value is not specified, the cache service default (0) is used.
Range: -1 to 2147483647

`--periodic-schedule=<period-spec>`

Specify a periodic schedule for refreshing the cached directory data. Use the following syntax:

```
--periodic-schedule="<number-of-units>
<unit-type>"
```

`<number-of-units>`	An integer from 1 to 2147483648
`<unit-type>`	Seconds, minutes, hours, days, weeks, or months

`--cron-schedule=<cron-spec>`

Create a a cron schedule that tells the cache service how often to refresh cached directory data. Use the following syntax:

```
--cron-schedule="<seconds> <minutes> <hours>
<days-of-month> <months> <days-of-week>
[<years>]"
```

For details about the <cron-spec> syntax, see Appendix C, “Configuring cron schedules”.

`--one-time-schedule=<time-stamp>`

Specify a single time and date for refreshing the cached directory data. Use the following syntax:

```
--one-time-schedule="{<hh>:<mm>:<ss> {am | pm}
<mm>-<dd>-<yyyy> | now}"
```

`--start-time=<time-stamp>`

Specify the time and date that the cache service will start refreshing cached directory data, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> |
now}
```

`--end-time=<time-stamp>` Specify the time and date that the cache service will stop refreshing cached directory data, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> | now}
```

`--max-iterations=<num-iterations>` Specify the number of times a schedule will repeat.

`<num-iterations>`	An integer from 1 to 2147483648. If no value is specified, the schedule repeats indefinitely.

`--time-zone=<time-zone>` Specify the time zone for the schedule. The schedule can be specified according to the cache server's time zone or relative to Greenwich Mean Time (GMT). If no value is specified for the time zone, the server's local time zone is used. The format for the time zone is as follows:

GMT+12:00 Fiji	GMT+02:00 Cairo
GMT+11:00 Noumea	GMT+01:00 Paris
GMT+10:00 Sydney	GMT London
GMT+9:30 Adelaide	GMT-01:00 Azores
GMT+09:00 Tokyo	GMT-02:00 Mid-Atlantic
GMT+08:00 Hong Kong	GMT-03:00 Rio de Janeiro
GMT+07:00 Bangkok	GMT-03:30 Newfoundland
GMT+06:30 Rangoon	GMT-04:00 Caracas
GMT+06:00 Dacca	GMT-05:00 New York
GMT+05:45 Katmandu	GMT-06:00 Chicago
GMT+05:30 Calcutta	GMT-07:00 Denver
GMT+05:00 Karachi	GMT-08:00 San Francisco
GMT+04:30 Kabul	GMT-09:00 Juneau
GMT+04:00 Baku	GMT-10:00 Hawaii
GMT+03:30 Tehran	GMT-11:00 Samoa
GMT+03:00 Moscow	

`--exclusion=<exclusion-name>`

The name of an exclusion to this schedule. For information about creating schedule exclusions, see **avaki scheduleexclusion --create --daily** (page 201), **avaki scheduleexclusion --create --weekly** (page 205), **avaki scheduleexclusion --create --monthly** (page 203), **avaki scheduleexclusion --create --yearly** (page 207), and **avaki scheduleexclusion --create --custom** (page 199).

`--exclusion-action={drop | sooner | later}`

Specify the action to take if the cache service is scheduled to refresh cached directory data during an exclusionary period.

drop	Don't refresh the cached directory data during the exclusionary period.
sooner	Reschedule the cache service to refresh the cached directory before the exclusionary period.
later	Reschedule the cache service to refresh the cached directory data after the exclusionary period.

`--reschedule-offset=<offset-spec>`

Specify the amount of time before or after the exclusionary period to perform a rescheduled cache refresh. Use the following syntax:

```
--reschedule-offset="<number-of-units>
<unit-type>"
```

`<number-of-units>`	An integer from 1 to 2147483648
`<unit-type>`	Seconds, minutes, hours, days, weeks, or months

`<cache-service-path>`

The grid path of a cache service.

`<directory-path>+`

The grid path of one or more directories to cache.

avaki directory --delete-schedule

Delete a caching schedule for a directory.

To use this command, you must have write permission on the cache service.

Syntax

```
avaki directory --delete-schedule <cache-service-path>
  <directory-path> <schedule-id>
```

`<cache-service-path>`	The grid path of a cache service.
`<directory-path>`	The grid path of the directory whose caching schedule will be deleted.
`<schedule-id>`	The numeric ID of the schedule to delete. To determine a schedule's ID, use **avaki directory --list-schedules** (page 127).

avaki directory --do-not-cache

Mark a cached directory for no caching.

Syntax

```
avaki directory --do-not-cache <cache-service-path>
  <directory-path>+
```

`<cache-service-path>`	The data catalog path of a cache service.
`<directory-path>`	The data catalog path of the directory to be marked for no caching.

avaki directory --list-schedules

List the schedules for a cached directory.

Syntax

```
avaki directory --list-schedules <cache-service-path>
  <directory-path>
```

`<cache-service-path>`	The data catalog path of a cache service.
`<directory-path>`	The data catalog path of the directory whose schedule will be listed.

avaki domain --create

Create an Avaki domain.

Syntax

```
avaki domain --create <domain> [<host>] [<gdc-port>] | <host>:
  <gdc-port>]
```

`<domain>`	Specify the name of the Avaki domain to create. The name can be up to 30 characters long.
`<host>`	Optional. The DNS name or IP address of the machine on which you want to create the new domain's GDC.
`<gdc-port>`	Optional. The connect port for the new grid domain controller. (The GDC uses the connect port to communicate with other Avaki objects; see the connect port entry in the Glossary, page 310, for more information.) If you omit this option, the system assumes that the connect port is 3099.

avaki domain --disconnect

Disconnect an Avaki domain from another domain.

Syntax

```
avaki domain --disconnect <other-domain>
```

`<other-domain>`	The name of the Avaki domain to disconnect from.

avaki domain --info

View a description of an Avaki domain, including lists of all servers in the domain and any connected domains.

Syntax

```
avaki domain --info
```

–	This command has no options.

avaki domain --interconnect

Connect an Avaki domain to another Avaki domain.

Syntax

```
avaki domain --interconnect <other-domain> <machine>:<port>
```

`<other-domain>`	The name of the Avaki domain to connect with.
`<machine>:<port>`	For <machine>, specify the DNS name or IP address of the GDC machine in the other Avaki domain. For port, specify the remote GDC machine's connect port number.

avaki executionservice --info

Display information about an execution service, including its name, the maximum number of concurrent data services, and any other grid servers whose execution services are pooled with this execution service.

Syntax

```
avaki executionservice --info <server-path>
```

`<server-path>`	The data catalog path to the Avaki server on which the execution service resides.

avaki executionservice --set

Configure an execution service.

Syntax

```
avaki executionservice --set
  [--max-concurrent=<maximum-data-services>]
  --pool=<server-path>* <server-path>
```

`--max-concurrent=<maximum-data-services>`	Specify an interval that represents the maximum number of concurrent data services.
`--pool=<server-path>*`	The path to one or more grid servers that the execution service is associated with. To specify more than one server, include the --pool argument for each server. The format is as follows: `--pool=<serverpath1> --pool=<serverpath2>`
`<server-path>`	The data catalog path to the grid server on which the execution service resides.

avaki file --add-schedule

Add a new pin or cache schedule to a currently cached file. (For information about tagging an Avaki directory or a directory hierarchy for caching within a cache service, see **avaki file --cache-on-demand** (page 135) and **avaki file --pin** (page 138)).

To use this command, you must have write permission on the cache service, and you must have read permission on the file.

Syntax

```
avaki file --add-schedule {--periodic-schedule=<period-spec> |
  --cron-schedule=<cron-spec> |
  --one-time-schedule=<time-stamp>}
  [--start-time=<time-stamp>] [[--end-time=<time-stamp>] |
  [--max-iterations=<num-iterations>]]
  [--time-zone=<time-zone>] [--exclusion=<exclusion-name>]+
  [--exclusion-action={ drop | sooner | later }]
  [--reschedule-offset=<offset-spec>] <cache-service-path>
  <file-path>+
```

`--periodic-schedule=<period-spec>`

Specify a periodic schedule for refreshing the cached file data. Use the following syntax:

```
--periodic-schedule="<number-of-units> <unit-type>"
```

`<number-of-units>`	An integer from 1 to 2147483648
`<unit-type>`	seconds, minutes, hours, days, weeks, or months

In this example, the periodic schedule is set to 30 seconds:

```
--periodic-schedule="30 seconds"
```

`--cron-schedule=<cron-spec>`

Specify a periodic schedule for refreshing the cached file data. Use the following syntax:

```
--cron-schedule="<seconds> <minutes> <hours>
<days-of-month> <months> <days-of-week>
[<years>]"
```

For details about the <cron-spec> syntax, see Appendix C, "Configuring cron schedules".

`--one-time-schedule=<time-stamp>`

Specify a single time and date for refreshing the cached file data. Use the following syntax:

```
--one-time-schedule="{<hh>:<mm>:<ss> {am | pm}
<mm>-<dd>-<yyyy> | now}"
```

`--start-time=<time-stamp>`

Specify the time and date that the cache service will start refreshing cached file data, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> |
now}
```

`--end-time=<time-stamp>`

Specify the time and date that the cache service will stop refreshing cached file data, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> |
now}
```

`--max-iterations=<num-iterations>`

Specify the number of times a schedule will repeat.

`<num-iterations>`	An integer from 1 to 2147483648. If no value is specified, the schedule repeats indefinitely.

`--time-zone=<time-zone>`

Specify the time zone for the schedule. The schedule can be specified according to the cache server's time zone or relative to Greenwich Mean Time (GMT). If no value is specified for the time zone, the server's local time zone is used. The format for the time zone is as follows:

GMT+12:00 Fiji	GMT+02:00 Cairo
GMT+11:00 Noumea	GMT+01:00 Paris
GMT+10:00 Sydney	GMT London

GMT+9:30 Adelaide	GMT-01:00 Azores
GMT+09:00 Tokyo	GMT-02:00 Mid-Atlantic
GMT+08:00 Hong Kong	GMT-03:00 Rio de Janeiro
GMT+07:00 Bangkok	GMT-03:30 Newfoundland
GMT+06:30 Rangoon	GMT-04:00 Caracas
GMT+06:00 Dacca	GMT-05:00 New York
GMT+05:45 Katmandu	GMT-06:00 Chicago
GMT+05:30 Calcutta	GMT-07:00 Denver
GMT+05:00 Karachi	GMT-08:00 San Francisco
GMT+04:30 Kabul	GMT-09:00 Juneau
GMT+04:00 Baku	GMT-10:00 Hawaii
GMT+03:30 Tehran	GMT-11:00 Samoa
GMT+03:00 Moscow	

`--exclusion=<exclusion-name>`

The name of an exclusion to this schedule. For information about creating schedule exclusions, see **avaki scheduleexclusion --create --daily** (page 201), **avaki scheduleexclusion --create --weekly** (page 205), **avaki scheduleexclusion --create --monthly** (page 203), **avaki scheduleexclusion --create --yearly** (page 207), and **avaki scheduleexclusion --create --custom** (page 199).

`--exclusion-action={drop | sooner | later}`

Specify the action to take if the cache service is scheduled to refresh cached file data during an exclusionary period.

drop	Don't refresh the cached file data during the exclusionary period.
sooner	Reschedule the cache service to refresh the cached file data before the exclusionary period.
later	Reschedule the cache service to refresh the cached file data after the exclusionary period.

`--reschedule-offset=<offset-spec>`	Specify the amount of time before or after the exclusionary period to perform a rescheduled cache refresh. Use the following syntax: `--reschedule-offset="<number-of-units> <unit-type>"` `<number-of-units>` — An integer from 1 to 2147483648 `<unit-type>` — Seconds, minutes, hours, days, weeks, or months
`<cache-service-path>`	The data catalog path of a cache service.
`<file-path>+`	The data catalog path of the file to which a pin schedule will be added.

Example

This example pins the file fredsform.txt for scheduled caching. The cached directory data is refreshed every 30 seconds and the time zone is GMT-05:00 New York.

```
$ avaki file --add-schedule --periodic-schedule="30 seconds"
--time-zone="GMT-05:00 New York" /System/LocalDomain/
Services/CacheServices/MAGADIEUXP.sybase.com "/Shares/Water
Buffalo Lodge/fredsform.txt"
```

avaki file --cache-on-demand

Tags a file within a cache service for on-demand caching.

Syntax

```
avaki file --cache-on-demand [--coherence-window=<value>]
  [--offline-expiration=value] <cache-service-path>
  <file-path>+
```

`--coherence-window=<value>`	Optional. The duration (in seconds) during which the cache service assumes the cached file content to be fresh after it last inspected the back-end source file for updates. The valid values are -1 and 0 to 2147483647 seconds. If the coherence window is not specified, the cache service default, 86400 (1 day),. is used. Set this attribute to -1 when the cache includes static files that should never expire, or set it to 0 to indicate that they should never be cached.
`--offline-expiration=<value>`	Optional. If the back-end file is offline, the cache service continues to return the cached data for the file for the duration (in seconds) specified by the offline-expiration value. A value of 0 indicates that the cache service is not allowed to return cached data after it has determined that the source is offline. A value of -1 indicates that the cached data never expires when the back-end source is offline. If the offline-expiration value is not specified, the cache service default (0) is used. Range: -1 to 2147483647
`<cache-service-path>`	The data catalog path of a cache service.
`<file-path>+`	The data catalog path of one or more files to tag for on-demand caching.

avaki file --delete-schedule

Delete a pin schedule from a cached file.

To use this command, you must have write permission on the cache service.

Syntax

```
avaki file --delete-schedule <cache-service-path> <file-path>
  <schedule-id>
```

`<cache-service-path>`	The data catalog path of a cache service.
`<file-path>`	The data catalog path of the file whose pin schedule will be deleted.
`<schedule-id>`	The numeric ID of the schedule to delete. To determine a schedule's ID, use **avaki file --list-schedules** (page 137).

avaki file --do-not-cache

Mark a file for no caching.

Syntax

```
avaki file --do-not-cache <cache-service-path> <file-path>+
```

`<cache-service-path>`	The data catalog path of a cache service.
`<file-path>+`	The data catalog path of one or more files to be marked for no caching.

avaki file --list-schedules

List the schedules for a cached file.

Syntax

```
avaki file --list-schedules <cache-service-path> <file-path>
```

`<cache-service-path>`	The data catalog path of a cache service.
`<file-path>`	The data catalog path of the file whose schedule will be listed.

avaki file --pin

Marks ("pins") a file within a cache service for scheduled caching.

Syntax

```
avaki file --pin [--offline-expiration=<value>]
  [{--periodic-schedule=<period-spec> |
  --cron-schedule=<cron-spec> |
  --one-time-schedule=<time-stamp>}]
  [--start-time=<time-stamp>] [[--end-time=<time-stamp>] |
  [--max-iterations=<num-iterations>]]
  [--time-zone=<time-zone>] [--exclusion=<exclusion-name>]+
  [--exclusion-action={ drop | sooner | later }]
  [--reschedule-offset=<offset-spec>] <cache-service-path>
  <file-path>+
```

`--offline-expiration=<value>`

Optional. If the back-end source file is offline, the cache service continues to return the cached data for the file for the duration (in seconds) specified by the offline-expiration value. A value of 0 indicates that the cache service is not allowed to return cached data after it has determined that the source is offline. A value of -1 indicates that the cached data never expires when the back-end source is offline. If the offline-expiration value is not specified, the cache service default (0) is used.
Range: -1 to 2147483647

`--periodic-schedule=<period-spec>`

Specify a periodic schedule for refreshing the cached file data. Use the following syntax:

```
--periodic-schedule="<number-of-units>
<unit-type>"
```

`<number-of-units>`	An integer from 1 to 2147483648
`<unit-type>`	Seconds, minutes, hours, days, weeks, or months

`--cron-schedule=<cron-spec>`

Specify a periodic schedule for refreshing the cached file data. Use the following syntax:

```
--cron-schedule="<seconds> <minutes> <hours>
<days-of-month> <months> <days-of-week>
[<years>]"
```

For details about the <cron-spec> syntax, see Appendix C, "Configuring cron schedules".

`--one-time-schedule=<time-stamp>`

Specify a single time and date for refreshing the cached file data. Use the following syntax:

```
--one-time-schedule="{<hh>:<mm>:<ss> {am | pm}
<mm>-<dd>-<yyyy> | now}"
```

`--start-time=<time-stamp>`

Specify the time and date that the cache service will start refreshing cached file data, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> | now}
```

`--end-time=<time-stamp>`

Specify the time and date that the cache service will stop refreshing cached file data, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> | now}
```

`--max-iterations=<num-iterations>`

Specify the number of times a schedule will repeat.

`<num-iterations>`	An integer from 1 to 2147483648. If no value is specified, the schedule repeats indefinitely.

`--time-zone=<time-zone>`

Specify the time zone for the schedule. The schedule can be specified according to the cache server's time zone or relative to Greenwich Mean Time (GMT). If no value is specified for the time zone, the server's local time zone is used. The format for the time zone is as follows:

GMT+12:00 Fiji	GMT+02:00 Cairo
GMT+11:00 Noumea	GMT+01:00 Paris
GMT+10:00 Sydney	GMT London
GMT+9:30 Adelaide	GMT-01:00 Azores

GMT+09:00 Tokyo	GMT-02:00 Mid-Atlantic
GMT+08:00 Hong Kong	GMT-03:00 Rio de Janeiro
GMT+07:00 Bangkok	GMT-03:30 Newfoundland
GMT+06:30 Rangoon	GMT-04:00 Caracas
GMT+06:00 Dacca	GMT-05:00 New York
GMT+05:45 Katmandu	GMT-06:00 Chicago
GMT+05:30 Calcutta	GMT-07:00 Denver
GMT+05:00 Karachi	GMT-08:00 San Francisco
GMT+04:30 Kabul	GMT-09:00 Juneau
GMT+04:00 Baku	GMT-10:00 Hawaii
GMT+03:30 Tehran	GMT-11:00 Samoa
GMT+03:00 Moscow	

`--exclusion=<exclusion-name>`

The name of an exclusion to this schedule. For information about creating schedule exclusions, see **avaki scheduleexclusion --create --daily** (page 201), **avaki scheduleexclusion --create --weekly** (page 205), **avaki scheduleexclusion --create --monthly** (page 203), **avaki scheduleexclusion --create --yearly** (page 207), and **avaki scheduleexclusion --create --custom** (page 199).

`--exclusion-action={drop | sooner | later}`

Specify the action to take if the cache service is scheduled to refresh cached file data during an exclusionary period.

drop	Don’t refresh the cached file data during the exclusionary period.
sooner	Reschedule the cache service to refresh the cached file data before the exclusionary period.
later	Reschedule the cache service to refresh the cached file data after the exclusionary period.

`--reschedule-offset=<offset-spec>`	Specify the amount of time before or after the exclusionary period to perform a rescheduled cache refresh. Use the following syntax: `--reschedule-offset="<number-of-units> <unit-type>"` `<number-of-units>` An integer from 1 to 2147483648 `<unit-type>` Seconds, minutes, hours, days, weeks, or months
`<cache-service-path>`	The data catalog path of a cache service.
`<file-path>+`	The grid paths of one or more files to tag for scheduled caching.

avaki group --add --user

Add a user to a group.

Syntax

```
avaki group {--add | -a} {--user | -u} {<qualified-user-name>}
  {<group-grid-path> | <qualified-group-name>}
```

`<qualified-user-name>`	Specify the qualified name of the user that you are adding to the group. The format is:
	`<user>@<authservice>.<authservicetype>.<domain>`
	For example:
	`wilma@DefaultAuthService.Grid.Bedrock`
	The elements of <qualified-user-name> are as follows:
	`<user>` The name of the user to add to the group.
	`<authservice>` The name of the user's authentication service.

	<authservicetype>	The type of authentication service (Grid, Nis, or Ldap).
	<domain>	The name of the user's Avaki domain.

<group-grid-path> | <qualified-group-name>

Specify the group to which you are adding the user. You can specify the data catalog path to the group directory. For example:

```
/System/Domains/Bedrock/Services/AuthSer-
vices/Grid/DefaultAuthService/Groups/Development
```

Alternatively, you can specify the qualified group name using the following format:

```
<group>@<authservice>.<authservicetype>.<domain>
```

For example:

```
Development@DefaultAthService.Grid.Bedrock
```

The elements of <qualified-group-name> are as follows:

<group>	The name of the group to which you are adding the user.
<authservice>	The name of the group's authentication service.
<authservicetype>	The type of the group's authentication service (Grid, Nis, or Ldap).
<domain>	The name of the group's Avaki domain.

avaki group --create

Create a new grid group.

Syntax

```
avaki group --create [--description=<description-text>]
  <grid-group>+
```

`--description=<description-text>`	Optional. Enter details about this group; enclose the entire text string in quotation marks. The description is shown to web UI users who display information about the group. Consider including an explanation of what this group is for or where to get more information. **Note:** If you create or import multiple groups with one command, the same description is applied to every group. Length limit: 255 characters
`<grid-group>+`	The name(s) of the group or groups to create.

avaki group --delete

Delete a group.

Note To list the groups in a authentication service, enter a command of the form `avaki ls <auth-service-directory>/Groups.` For example, **avaki ls /System/Domains/Bedrock/Services/AuthServices/Ldap/testldap/Groups**.

Syntax

```
avaki group {--delete | -d} <auth-service-path> {{--all | -a} |
  <group-name>+}
```

`<auth-service-path>`	The data catalog path to the authentication service from which you are deleting the group. For example: /System/LocalDomain/Services/AuthServices/Grid/Default-AuthService/AuthService
`--all | -a`	Delete all groups from the specified authentication service.
`<group-name>`	The name of the group you are deleting. To specify more than one group, enter a space-delimited list of group names.For example: `Development QA`

avaki group --delete --user

Delete a user from a group.

Syntax

```
avaki group {--delete | -d} {--user | -u}
  {<qualified-user-name>} {<group-grid-path> |
  <qualified-group-name>}
```

`<qualified-user-name>`

Specify the qualified name of the user that you are deleting from the group. The format is:

```
<user>@<authservice>.<authservice-type>.<domain>
```

For example:

```
wilma@DefaultAuthService.Grid.Bedrock
```

The elements of <qualified-user-name> are as follows:

`<user>`	The name of the user to delete from the group.
`<authservice>`	The name of the user's authentication service.
`<authservicetype>`	The type of authentication service (Grid, Nis, or Ldap).
`<domain>`	The name of the user's Avaki domain.

`<group-grid-path> | <qualified-group-name>`

Specify the qualified name of the group from which you are deleting the user. The format is:

```
<group>@<authservice>.<authservicetype>.<domain>
```

For example:

```
Development@DefaultAuthService.Grid.Bedrock
```

The elements of <qualified-group-name> are as follows:

`<group>`	The path or name of the group from which you are removing the user.
`<authservice>`	The name of the group's authentication service.
`<authservicetype>`	The type of the group's authentication service (Grid, Nis, or Ldap).
`<domain>`	The name of the group's Avaki domain.

avaki group --info

Display a group's description.

Syntax

```
avaki group --info {<group-grid-path> |
  <qualified-group-name>}}
```

`<group-grid-path> | <qualified-group-name>`

The name of the group that you are obtaining information about.

You can specify the data catalog path to the directory of the group. For example:

```
/System/Domains/Bedrock/Services/AuthSer-
vices/Grid/DefaultAuthService/Groups/Develop-
ment
```

Alternatively, you can specify the qualified group name using the following format:

```
<group>@<authservice>.<authservice-
type>.<domain>
```

For example:

```
Development@DefaultAthService.Grid.Bedrock
```

The elements of <qualified-group-name> are as follows:

`<group>`	The name of the group for which you are obtaining information.
`<authser-vice>`	The name of the group's authentication service.
`<authser-vicetype>`	The type of the group's authentication service (Grid, Nis, or Ldap).
`<domain>`	The name of the group's Avaki domain.

avaki group --list-user

List all users in a group.

Syntax

```
avaki group {--list-user | -l} {<group-grid-path> |
  <qualified-group-name>}
```

`<group-grid-path> | <qualified-group-name>`

The grid path of the group for which you want to obtain a list of users. You can specify the path to the grid directory of the group. For example:

```
/System/LocalDomain/Services/AuthSer-
vices/Grid/DefaultAuthService/Groups/Development
```

Alternatively, you can specify the qualified group name by using the following format:

```
<group>@<authservice>.<authservicetype>.
<domain>
```

For example:

```
Development@DefaultAthService.Grid.Bedrock
```

The elements of <qualified-group-name> are as follows:

`<group>`	The name of the group for which you want to obtain a list of users.
`<authservice>`	The name of the group's authentication service.
`<authservice-type>`	The type of the group's authentication service (Grid, Nis, or Ldap).
`<domain>`	The name of the group's Avaki domain.

avaki help

View a list of Avaki commands and the Avaki release version number, or view usage information for a particular command.

Syntax

```
avaki help [<cmd>] [--verbose | -v]
```

`<cmd>`	Optional. The command for which you want to view usage information. If no command is specified, the system displays a list of all Avaki commands.
`--verbose | -v`	Optional. Display the commands' syntax and descriptions.

avaki id

Download an interconnection ID for a user or group.

Syntax

```
avaki id --user={<full-user-path> | <full-group-path>}
  [--filename=<output-filename>]
```

`--user= {<full-user-path> | <full-group-path>}`	The grid directory of the user or group for which you want to obtain an interconnection ID.
`--filename= <output-filename>`	Optional. Specify a filename and path in the local file system for the interconnection ID that this command generates.

avaki ldap --add-schedule

Add a schedule according to which users will be imported automatically. When users are imported according to a schedule, the list of currently imported users is refreshed and any new users are imported automatically.

To use this command, you need write permission on the LDAP authentication service.

Note If you want to import different groups on different schedules, configure a cron specification for the avaki ldap --add-schedule command and use it in conjunction with **avaki ldap --import** (page 155).

Syntax

```
avaki ldap --add-schedule <auth-service>
  {--periodic-schedule=<period-spec> |
  --cron-schedule=<cron-spec> |
  --one-time-schedule=<time-stamp>}
  [--start-time=<time-stamp>] [[--end-time=<time-stamp>] |
  [--max-iterations=<num-iterations>]]
  [--time-zone=<time-zone>] [--exclusion=<exclusion-name>]+
  [--exclusion-action={ drop | sooner | later }]
  [--reschedule-offset=<offset-spec>]
```

`<auth-service>`	The name of the authentication service for which the schedule will be added.
`--periodic-schedule=<period-spec>`	Specify a periodic schedule for importing users automatically. Use the following syntax: `--periodic-schedule="<number-of-units> <unit-type>"` `<number-of-units>` — An integer from 1 to 2147483648 `<unit-type>` — seconds, minutes, hours, days, weeks, or months

In the following example, the periodic schedule is set to 30 seconds:

```
--periodic-schedule="30 seconds"
```

`--cron-schedule=<cron-spec>`

Specify how often a cron-based scheduler for importing users will import users. Use the following syntax:

```
--cron-schedule="<seconds> <minutes> <hours>
<days-of-month> <months> <days-of-week>
[<years>]"
```

For details about the <cron-spec> syntax, see Appendix C, "Configuring cron schedules".

`--one-time-schedule=<time-stamp>`

Specify a single time and date for importing the users. Use the following syntax:

```
--one-time-schedule="{<hh>:<mm>:<ss> {am | pm}
<mm>-<dd>-<yyyy> | now}"
```

`--start-time=<time-stamp>`

Specify the time and date that the users will be imported, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> |
now}
```

`--end-time=<time-stamp>`

Specify the time and date that the users will be imported, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> |
now}
```

`--max-iterations=<num-iterations>`

Specify the number of times a schedule will repeat.

`<num-iterations>`	An integer from 1 to 2147483648. If no value is specified, the schedule repeats indefinitely.

`--time-zone=<time-zone>`

Specify the time zone for the schedule. The schedule can be specified according to the grid server's time zone or relative to Greenwich Mean Time (GMT). If no value is specified for the time zone, the server's local time zone is used. The format for the time zone is as follows:

GMT+12:00 Fiji	GMT+02:00 Cairo
GMT+11:00 Noumea	GMT+01:00 Paris
GMT+10:00 Sydney	GMT London
GMT+9:30 Adelaide	GMT-01:00 Azores
GMT+09:00 Tokyo	GMT-02:00 Mid-Atlantic
GMT+08:00 Hong Kong	GMT-03:00 Rio de Janeiro
GMT+07:00 Bangkok	GMT-03:30 Newfoundland
GMT+06:30 Rangoon	GMT-04:00 Caracas
GMT+06:00 Dacca	GMT-05:00 New York
GMT+05:45 Katmandu	GMT-06:00 Chicago
GMT+05:30 Calcutta	GMT-07:00 Denver
GMT+05:00 Karachi	GMT-08:00 San Francisco
GMT+04:30 Kabul	GMT-09:00 Juneau
GMT+04:00 Baku	GMT-10:00 Hawaii
GMT+03:30 Tehran	GMT-11:00 Samoa
GMT+03:00 Moscow	

`--exclusion=<exclusion-name>`

The name of an exclusion to this schedule. For information about creating schedule exclusions, see **avaki scheduleexclusion --create --daily** (page 201), **avaki scheduleexclusion --create --weekly** (page 205), **avaki scheduleexclusion --create --monthly** (page 203), **avaki scheduleexclusion --create --yearly** (page 207), and **avaki scheduleexclusion --create --custom** (page 199).

`--exclusion-action={drop | sooner | later}`

Specify the action to take if the authentication service is scheduled to update users automatically during an exclusionary period.

drop	Don't run the import task during the exclusionary period.
sooner	Reschedule the import task to run before the exclusionary period.
later	Reschedule the import task to run after the exclusionary period.

`--reschedule-offset=<offset-spec>`

Specify the amount of time before or after the exclusionary period to perform a user import. Use the following syntax:

```
--reschedule-offset="<number-of-units> <unit-type>"
```

`<number-of-units>`	An integer from 1 to 2147483648
`<unit-type>`	Seconds, minutes, hours, days, weeks, or months

Example

This example add a schedule in which users will be imported automatically into the MyLDAP authentication service every 2 hours. In this example, the time zone is GMT-05:00 New York.

```
$ avaki ldap --add-schedule MyLDAP --periodic-schedule=
"2 hours" --time-zone="GMT-05:00 New York"
```

avaki ldap --delete

Delete an LDAP authentication service or delete users from an authentication service.

Syntax

To delete an authentication service:

```
avaki ldap {--delete | -d} <auth-service>
```

To delete users:

```
avaki ldap {--delete | -d} {--user | -u}
  {<user-name>+ | {--all | -a}} <auth-service>
```

`<auth-service>`	The name of the authentication service to delete.
`--user | -u`	Delete one or more users from the specified authentication service.
`<user-name>+`	The name or names of user accounts to import. To specify more than one user, enter a space-delimited list of user IDs.
`--all | -a`	Import all users or groups from the specified authentication service.

avaki ldap --delete-schedule

Delete a schedule that specifies when users will be imported automatically.

To use this command, you must have write permission on the LDAP authentication service.

Syntax

```
avaki ldap --delete-schedule <auth-service> <schedule-id>
```

`<auth-service>`	The name of the authentication service for which the schedule will be deleted.
`<schedule-id>`	The numeric ID of the schedule to delete. To determine a schedule's ID, use **avaki ldap --list-schedules** (page 160).

avaki ldap --import

Import user accounts or groups from an LDAP authentication service into an Avaki grid domain.

Note When you import a user, you import all the groups to which the user belongs. When you import a group, you import all users in the group and all groups to which those users belong. Exceptions are related to the Domain Users group in Active Directory: importing an AD Domain Users group has no effect. And when you import a member of an AD Domain Users group, the Domain Users group is not imported into the grid domain. (Other groups that the user belongs to will be imported as usual.)

The import uses the searchbase you entered when you integrated the LDAP directory service.

Syntax

To import user accounts:

```
avaki ldap {--import | -i} {--user | -u} <user-name>+
  <auth-service>

avaki ldap {--import | -i} {--user | -u} --dn <user-name>+
  <auth-service>

avaki ldap {--import | -i} {--user | -u} {--all | -a}
  <auth-service>
```

To import groups:

```
avaki ldap {--import | -i} {--group | -g} {<group-name>+ |
  [--all | -a]} <auth-service>
```

`--user | -u`	Import users.
`<user-name>+`	The name or names of user accounts to import. To specify more than one user, enter a space-delimited list of user IDs. If you use the --dn option, each name must be an LDAP distinguished name (for example, CN=Joe Smith, OU=Burlington, DC=Avaki, DC=com). Enclose each DN string in quotation marks and separate the strings with spaces.
`--dn`	Indicates that you are providing LDAP distinguished names in the <users> argument.
`--all | -a`	Import all users or groups from the specified authentication service.
`--group | -g`	Import groups.
`<group-name>+`	The name or names of the groups to import. To specify more than one group, enter a space-delimited list of group names.
`<auth-service>`	The name of the authentication service to import.

Examples

This command uses LDAP distinguished names to import user accounts for Fred Flintstone and Barney Rubble from the testldap authentication service.

```
$ avaki ldap --import -u --dn "CN=Fred Flintstone,
OU=Bedrock, DC=Slate Quarry, DC=com" "CN=Barney Rubble,
OU=Bedrock, DC=Slate Quarry, DC=com" testldap
```

This command imports all groups from the testldap authentication service.

```
$ avaki ldap --import -g -a testldap
```

avaki ldap --info

Display information about an LDAP authentication service.

Syntax

```
avaki ldap --info <auth-service>
```

`<auth-service>`	The name of the authentication service from which to add or delete the search base.

avaki ldap --integrate

Integrate an Active Directory or other LDAP-compliant authentication service into an Avaki domain.

Syntax

```
avaki ldap --integrate <auth-service>
  {--host | -h}=<ldap-host>[:<port-number>]
  --auth-type=<type> --auth-dn=<search-base>
  --auth-pw=<password> --user-searchbase=<search-base>
  --group-searchbase=<search-base>
  --user-account-attr=<user-name-attribute-name>
  --group-account-attr=<group-name-attribute-name>
  [--uid-attr=<uid-attribute-name>]
  [--refresh-on-login] [--enabled-query=<query-string>]
```

`<auth-service>`	Devise a name for the new service.
`{--host | -h}=<ldap-host>[:<port-number>]`	The name or IP address of the host machine on which the LDAP authentication service resides. Optionally, you can include the port number on the LDAP host. By default, Avaki uses the default LDAP host port, 389. If your LDAP host uses another port, use --host or -h to specify it.
`--auth-type=<type>`	The authentication server's type. For <type>, specify simple.
`--auth-dn=<search-base>`	Enter one or more LDAP name components in the Search Base field. The string you enter here must include values for CN, OU, DC—for example, CN=Joe Smith, OU=Burlington, DC=Avaki, DC=com. If you do not know what to enter here, consult your LDAP administrator.
`--auth-pw=<password>`	The password for the authentication server. If you're executing this command on a Unix system and the password contains non-alphanumeric characters (! or &, for instance), enclose the password in single quotation marks to prevent the shell from interpreting the special characters.

`--user-searchbase=<search-base>`	The top level of the first LDAP directory context within which the system will search for users. For example, you might enter OU=Burlington, DC=Avaki, DC=com. To specify more than one searchbase, use commas to separate the searchbase names. The system searches on the LDAP name components to find users that you can import. If you do not know what to enter here, consult your LDAP administrator.
`--group-searchbase=<search-base>`	The top level of the first LDAP directory context within which the system will search for groups. For example, you might enter OU=Groups, DC=Avaki, DC=com. To specify more than one searchbase, use commas to separate the searchbase names. The system searches on the LDAP name components to find groups that you can import. If you do not know what to enter here, consult your LDAP administrator.
`--user-account-attr=<user-name-attribute-name>`	Enter the name of the LDAP attribute that contains the username that a user from this authentication service will use to log in to this Avaki domain. For example, enter "sAMAccountName" for an out-of-the-box Active Directory installation, or "uid" for an out-of-the-box iPlanet installation.
`--group-account-attr=<group-name-attribute-name>`	Enter the name of the LDAP attribute that contains the group name that users from this authentication service will use to log in to this Avaki domain. For example, enter "cn" for out-of-the-box Active Directory and iPlanet installations.
`--uid-attr=<uid-attribute-name>`	Optional. Enter the name of an LDAP attribute that contains UID information.
`[--refresh-on-login]`	Optional. Refresh a user's information (such as group memberships) from the external directory each time the user logs in.
`[--enabled-query=<query-string>`	Optional. Specify a query that will be sent to the LDAP server to determine whether a user is enabled. For information about the format for LDAP queries, see your LDAP documentation.

avaki ldap --list-schedules

Display a list of schedules that specify when users will be imported automatically.

Syntax

```
avaki ldap --list-schedules <auth-service>
```

`<auth-service>`	The name of the authentication service for which the schedule will be displayed.

avaki ldap --searchbase

Add or delete a search base for an Active Directory or other LDAP-compliant authentication service that has been integrated into an Avaki domain.

Syntax

```
avaki ldap --searchbase {--add | --delete} {{--user | -u} |
  {--group | -g}} <auth-service> <search-base>
```

`--add | --delete`	Specify --add to add a search base or --delete to delete one.
`--user | -u`	Specify --user or -u to add or delete a user search base.
`--group | -g`	Specify --group or -g to add or delete a group search base.
`<auth-service>`	The name of the authentication service from which to add or delete the search base.
`<search-base>`	The LDAP directory context within which to add or delete the search base. For example, you might enter OU=Burlington, DC=Avaki, DC=com. If you do not know what to enter here, consult your LDAP administrator.

avaki ldap --update

Update an Active Directory or other LDAP-compliant authentication service that has been integrated into an Avaki domain.

Syntax

```
ldap --update <auth-service> {--host | -h}=<ldap-host>
  --auth-type=<type> --auth-dn=<search-base>
  --auth-pw=<password> --user-searchbase=<search-base>
  --group-searchbase=<search-base>
  --user-account-attr=<user-name-attribute-name>
  --group-account-attr=<group-name-attribute-name>
  [--uid-attr=<uid-attribute-name>]
  [--enabled-query=<query-string>]
```

`<auth-service>`	The name of the authentication service.
`{--host | -h}=<ldap-host>`	The name or IP address of the host machine on which the LDAP authentication service resides.
`--auth-type=<type>`	The authentication server's type. For <type>, specify simple.
`--auth-dn=<search-base>`	Enter one or more LDAP name components in the Search Base field. The string you enter here must include values for CN, OU, DC—for example, CN=Joe Smith, OU=Burlington, DC=Avaki, DC=com. If you do not know what to enter here, consult your LDAP administrator.
`--auth-pw=<password>`	The password for the authentication server. If you're executing this command on a Unix system and the password contains non-alphanumeric characters (! or &, for instance), enclose the password in single quotation marks to prevent the shell from interpreting the special characters.

`--user-searchbase=<search-base>`	The top level of the first LDAP directory context within which the system will search for users. For example, you might enter OU=Burlington, DC=Avaki, DC=com. To specify more than one searchbase, use commas to separate the searchbase names. The system searches on the LDAP name components to find users that you can import. If you do not know what to enter here, consult your LDAP administrator.
`--group-searchbase=<search-base>`	The top level of the first LDAP directory context within which the system will search for groups. For example, you might enter OU=Groups, DC=Avaki, DC=com. To specify more than one searchbase, use commas to separate the searchbase names. The system searches on the LDAP name components to find groups that you can import. If you do not know what to enter here, consult your LDAP administrator.
`--user-account-attr=<user-name-attribute-name>`	Enter the name of the LDAP attribute that contains the username that a user from this authentication service will use to log in to this Avaki grid domain. For example, enter "sAMAccountName" for an out-of-the-box Active Directory installation, or "uid" for an out-of-the-box iPlanet installation.
`--group-account-attr=<group-name-attribute-name>`	Enter the name of the LDAP attribute that contains the group name that users from this authentication service will use to log in to this Avaki grid domain. For example, enter "cn" for out-of-the-box Active Directory and iPlanet installations.
`--uid-attr=<uid-attribute-name>`	Optional. Enter the name of an LDAP attribute that contains UID information.
`[--enabled-query=<query-string>`	Optional. Specify a query that will be sent to the LDAP server to determine whether a user is enabled. For information about the format for LDAP queries, see your LDAP documentation.

avaki ln

Create a link between a grid file or directory and another location in the data catalog. Similar to the Unix **ln** command.

You can create *hard links* or *soft links*. A hard link (the default type) is an alternate name for a file or directory; changes to the object's other names have no effect on the hard link. A soft link is a pointer to a particular location (name) in the data catalog. If the object at that location is moved, deleted, or renamed, the soft link breaks. If you need to create a link to an Avaki generated view, use a soft link.

Syntax

```
avaki ln [-s] <target-grid-path> {<link-name> |
  <directory-target>}
```

`-s`	Soft link (optional).
`<target-grid-path>`	The path to the grid file or directory for which you are creating a link.
`<link-name>`	If you want to give a link a different name than the object to which it refers, specify a link name or the directory in which the link will be placed. The path may be relative or absolute.
`<directory-target>`	The directory in which the link will be placed. The link will have the same name as the object to which the link refers.

avaki locks --clear

Clear any file locks that are held on a file.

Syntax

```
avaki locks {--clear | -c} <grid-file>
```

`<grid-file>` The grid path of the locked file.

avaki locks --list

Lists the file locks that are currently held on a file.

Syntax

```
avaki locks {--list | -l} <grid-file>
```

`<grid-file>` The grid path of the file for which you want lock information.

avaki login

Log in to an Avaki domain.

Syntax

```
avaki login {--user=<user> | <user>} {--password=<password> |
  <password>} [--auth-service=<auth-service>]
  [--type=<service-type>] [--domain=<domain>]
```

`--user=<user> | <user>`

Your Avaki user name. You can specify your user name by itself (for example, wilma), or you can specify your qualified user name. The format for the qualified user name is as follows:

```
<user>@<authservice>.<authservice-type>.<domain>
```

For example:

```
wilma@DefaultAuthService.Grid.Bedrock
```

If you specify a qualified user name, do not specify <auth-service>, <service-type>, or <domain> separately.

The elements of <qualified-user-name> are as follows:

`<user>`	Your user name.
`<authservice>`	Your authentication service.
`<authservicetype>`	The type of authentication service (Grid, Nis, or Ldap).
`<domain>`	Your Avaki domain name.

`--password=<password> | <password>`

Your Avaki password. If you're executing this command on a Unix system and your password contains nonalphanumeric characters (! or &, for instance), enclose your password in single quotation marks to prevent the shell from interpreting the special characters.

`--auth-service =<auth-service>`	Optional. The name of your Avaki authentication service.
`--type= <service-type>`	Optional. Your authentication service type (grid, NIS, or LDAP).
`--domain=<domain>`	Optional. The name of the Avaki domain to log in to.

avaki logout

Log out of an Avaki domain.

Syntax

```
avaki logout
```

–	This command has no options.

avaki ls

Display a list of objects (such as files) in the specified grid directory, or if no directory is specified, list the objects in the current working directory. Similar to the Unix **ls** command.

Syntax

```
avaki ls [-r] [<grid-path>+]
```

`-r`	Recursive mode (optional). If you use the -r option, the system displays a list of objects in the specified grid directory and any of the directory's subdirectories.
`<grid-path>+`	Optional. One or more files or grid directories to be listed.

avaki mkdir

Create one or more grid directories. Similar to the Unix **mkdir** command.

Syntax

```
avaki mkdir [-p] [{--server | -s}=<grid-server>] <grid-path>+
```

`-p`	Optional. If the parent directory in <grid-path> doesn't exist, the parent directory will be created.
`{--server | -s}=<grid-server>`	Optional. Specify the name of the Avaki server on which the directory will be created. For example, Bedrock. If you omit this option, the directory will be created on the same server as the parent directory.
`<grid-path>+`	The grid path of the new directories to create.

avaki monitor --add

Add a monitor test to a monitor service.

Syntax

```
avaki monitor <monitor-name> --add --target=<target-grid-path>
  [--grid-server | --share-server | --proxy-server |
  --dgas-server --secondary-gdc-server] --type=<task>
  --retries=<retries> --period=<period-in-seconds>
```

`<monitor-name>`	The name of the monitor service to which the monitor test will be added.
`--target=<target-grid-path>`	The grid directory of the grid server or share server that you want to monitor.
`--grid-server | --share-server | --proxy-server | --dgas-server --secondary-gdc-server`	Optional. Specify the server to which the monitor test will be added. If you do not specify the server, you must use the --target option to specify full path of the object to monitor. If you do specify the server, make sure the --target option contains only the name of the object, not the full path to the object. --grid-server — Grid server --share-server — Share server --proxy server — Proxy server --dgas-server — Data grid access server --secondary-gdc-server — Secondary GDC
`--type=<task>`	The type of monitor test to set up.

	PingTask	This test sends one or more IP datagrams to a specified destination host and requests a reply, then measures the round-trip time.
	MessageTask	This test sends one or more JMS messages to a specified destination host and listens to a response message, then measures the round-trip time.
`--retries=<retries>`	The number of times the monitor service will attempt to repeat the test before declaring it a failure. (For example, 1 retry, the default, means that two consecutive pings must fail in order to generate a test failure in the log.)	
`--period=<period-in-secs>`	The time (in milliseconds) that the test waits for a response before declaring a failure.Timeout defaults to 10,000 (10 seconds).	

avaki monitor --clear

Remove tests from a monitor service. To remove one test, you must specify exactly the same options used when the test was added. (Use **avaki monitor --list** (page 172) to display the option values.)

Syntax

To remove all tests from the specified monitor service:

```
avaki monitor <monitor-name> --clear
```

To remove a particular test:

```
avaki monitor <monitor-name> --clear
  [--target=<target-grid-path> [--grid-server | --share-server
  | --proxy-server | --dgas-server --secondary-gdc-server]
  --type=<task> --retries=<retries>
  --period=<period-in-seconds>]
```

`<monitor-name>`	The name of the monitor service from which the monitor test will be removed.
`--target=<target-grid-path>`	The grid directory of the Avaki server whose monitor test you want to remove.
`--grid-server | --share-server | --proxy-server | --dgas-server --secondary-gdc-server`	Optional. Specify the server from which the monitor test will be removed.

--grid-server	Grid server
--share-server	Share server
--proxy server	Proxy server
--dgas-server	Data grid access server

	--secondary-gdc-server	Secondary GDC
`--type=<task>`	The type of monitor test to remove.	
	PingTask	This test sends one or more IP datagrams to a specified destination host and requests a reply, then measures the round-trip time.
	MessageTask	This test sends one or more JMS messages to a specified destination host and listens to a response message, then measures the round-trip time.
`--retries=<retries>`	The number of times the monitor service will attempt to repeat the test before declaring it a failure.	
`--period=<period-in-secs>`	The time (in milliseconds) that the test waits for a response before declaring a failure.Timeout defaults to 10,000 (10 seconds).	

avaki monitor --create

Create a monitor service.

Syntax

```
avaki monitor <monitor-name> --create
  [{--server | -s}=<grid-server>]
```

`<monitor-name>`	The name of the monitor service you are creating.
`--{server | -s}=<grid-server>`	Optional. Specify the name of the Avaki server on which the monitor service will be created. For example, Bedrock.

avaki monitor --delete

Delete a monitor service from the grid.

Syntax

```
avaki monitor <monitor-name> --delete
```

`<monitor-name>`	The name of the monitor service to delete.

avaki monitor --list

Display a list of all active monitor tests. The list includes details about each test: the test type, target grid path, number of retries, and period.

Syntax

```
avaki monitor <monitor-name> --list
```

`<monitor-name>`	The name of the monitor service from which to obtain a list of active tests.

avaki monitor --result

View the results of all tests of type <task> that have been added to a given monitor service.

Syntax

```
avaki monitor <monitor-name> --result --type=<task>
```

`<monitor-name>`	The name of the monitor service from which to obtain test results.	
`--type=<task>`	The type of monitor test to view.	
	PingTask	This test sends one or more IP datagrams to a specified destination host and requests a reply, then measures the round-trip time.
	MessageTask	This test sends one or more JMS messages to a specified destination host and listens to a response message, then measures the round-trip time.

avaki monitor --start

Restart all tests that have been added to a given monitor service.

Syntax

```
avaki monitor <monitor-name> --start
  [--target=<target-grid-path> [--grid-server | --share-server
  | --proxy-server | --dgas-server --secondary-gdc-server]
  --type=<task> --retries=<retries>
  --period=<period-in-seconds>]
```

`<monitor-name>`	The name of the monitor service on which monitor tests will be restarted.
`--target=<target-grid-path>`	The grid directory of the grid server or share server that you want to monitor.
`--grid-server | --share-server | --proxy-server | --dgas-server --secondary-gdc-server`	Optional. Specify the type of server on which the monitor test will be restarted.

--grid-server	Grid server
--share-server	Share server
--proxy server	Proxy server
--dgas-server	Data grid access server
--secondary-gdc-server	Secondary GDC

`--type=<task>`	The type of monitor test to restart.	
	PingTask	This test sends one or more IP datagrams to a specified destination host and requests a reply, then measures the round-trip time.
	MessageTask	This test sends one or more JMS messages to a specified destination host and listens to a response message, then measures the round-trip time.
`--retries=<retries>`	The number of times the monitor service will attempt to repeat the test before declaring it a failure.	
`--period=<period-in-secs>`	The time (in milliseconds) that the test waits for a response before declaring a failure.Timeout defaults to 10,000 (10 seconds).	

avaki monitor --stop

Temporarily stop all active tests that have been added to a given monitor service.

Syntax

```
avaki monitor <monitor-name> --stop
  [--target=<target-grid-path> [--grid-server | --share-server
  | --proxy-server | --dgas-server --secondary-gdc-server]
  --type=<task> --retries=<retries>
  --period=<period-in-seconds>]
```

`<monitor-name>`	The name of the monitor service on which monitor tests will be stopped.
`--target=<target-grid-path>`	The grid directory of the grid server or share server that you want to stop monitoring.
`--grid-server | --share-server | --proxy-server | --dgas-server --secondary-gdc-server`	Optional. Specify the type of server on which the monitor test will be stopped.

--grid-server	Grid server
--share-server	Share server
--proxy server	Proxy server
--dgas-server	Data grid access server
--secondary-gdc-server	Secondary GDC

`--type=<task>`	The type of monitor test to stop.

	PingTask	This test sends one or more IP datagrams to a specified destination host and requests a reply, then measures the round-trip time.
	MessageTask	This test sends one or more JMS messages to a specified destination host and listens to a response message, then measures the round-trip time.
`--retries=<retries>`		The number of times the monitor service will attempt to repeat the test before declaring it a failure.
`--period=<period-in-secs>`		The time (in milliseconds) that the test waits for a response before declaring a failure.Timeout defaults to 10,000 (10 seconds).

avaki mv

Move or rename an Avaki directory or a file in an Avaki directory. Similar to the Unix **mv** command.

Syntax

```
avaki mv [--force | -f] <source-grid-path> <target-grid-path>
```

`--force | -f`	If you specify this option, the directory or file will be moved even if the forcedDelete attribute for a directory or file is set to true. By default, the forcedDelete attribute is set to true for system-related directories and files that should be moved only with caution.
`<source-grid-path>`	The path to the grid directory or file that will be moved.
`<target-grid-path>`	The grid directory to which the file or grid directory will be moved. By default, the file or directory will retain its original name when moved. If you want to rename it, specify the new name in <target-grid-path>.

avaki nis --add-schedule

Add a schedule according to which users will be imported automatically.

To use this command, you need write permission on the NIS authentication service.

Syntax

```
avaki nis --add-schedule <auth-service>
  {--periodic-schedule=<period-spec> |
  --cron-schedule=<cron-spec> |
  --one-time-schedule=<time-stamp>}
  [--start-time=<time-stamp>] [[--end-time=<time-stamp>] |
  [--max-iterations=<num-iterations>]]
  [--time-zone=<time-zone>] [--exclusion=<exclusion-name>]+
  [--exclusion-action={ drop | sooner | later }]
  [--reschedule-offset=<offset-spec>]
```

`<auth-service>` — The name of the authentication service for which the schedule will be added.

`--periodic-schedule=<period-spec>` — Specify a periodic schedule for importing users automatically. Use the following syntax:

```
--periodic-schedule="<number-of-
units> <unit-type>"
```

`<number-of-units>`	An integer from 1 to 2147483648
`<unit-type>`	seconds, minutes, hours, days, weeks, or months

In the following example, the periodic schedule is set to 30 seconds:

```
--periodic-schedule="30 seconds"
```

`--cron-schedule=<cron-spec>`

Specify how often a a cron-based scheduler for importing users will import users. Use the following syntax:

```
--cron-schedule="<seconds> <minutes> <hours>
<days-of-month> <months> <days-of-week>
[<years>]"
```

For details about the <cron-spec> syntax, see Appendix C, "Configuring cron schedules".

`--one-time-schedule=<time-stamp>`

Specify a single time and date for importing the users. Use the following syntax:

```
--one-time-schedule="{<hh>:<mm>:<ss> {am | pm}
<mm>-<dd>-<yyyy> | now}"
```

`--start-time=<time-stamp>`

Specify the time and date that the users will be imported, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> |
now}
```

`--end-time=<time-stamp>`

Specify the time and date that the users will be imported, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> |
now}
```

`--max-iterations=<num-iterations>`

Specify the number of times a schedule will repeat.

`<num-iterations>`	An integer from 1 to 2147483648. If no value is specified, the schedule repeats indefinitely.

`--time-zone=<time-zone>`

Specify the time zone for the schedule. The schedule can be specified according to the grid server's time zone or relative to Greenwich Mean Time (GMT). If no value is specified for the time zone, the server's local time zone is used. The format for the time zone is as follows:

GMT+12:00 Fiji	GMT+02:00 Cairo
GMT+11:00 Noumea	GMT+01:00 Paris
GMT+10:00 Sydney	GMT London

GMT+9:30 Adelaide	GMT-01:00 Azores
GMT+09:00 Tokyo	GMT-02:00 Mid-Atlantic
GMT+08:00 Hong Kong	GMT-03:00 Rio de Janeiro
GMT+07:00 Bangkok	GMT-03:30 Newfoundland
GMT+06:30 Rangoon	GMT-04:00 Caracas
GMT+06:00 Dacca	GMT-05:00 New York
GMT+05:45 Katmandu	GMT-06:00 Chicago
GMT+05:30 Calcutta	GMT-07:00 Denver
GMT+05:00 Karachi	GMT-08:00 San Francisco
GMT+04:30 Kabul	GMT-09:00 Juneau
GMT+04:00 Baku	GMT-10:00 Hawaii
GMT+03:30 Tehran	GMT-11:00 Samoa
GMT+03:00 Moscow	

`--exclusion=<exclusion-name>`

The name of an exclusion to this schedule. For information about creating schedule exclusions, see **avaki scheduleexclusion --create --daily** (page 201), **avaki scheduleexclusion --create --weekly** (page 205), **avaki scheduleexclusion --create --monthly** (page 203), **avaki scheduleexclusion --create --yearly** (page 207), and **avaki scheduleexclusion --create --custom** (page 199).

`--exclusion-action={drop | sooner | later}`

Specify the action to take if the authentication service is scheduled to update users automatically during an exclusionary period.

drop	Don’t run the import task during the exclusionary period.
sooner	Reschedule the import task to run before the exclusionary period.
later	Reschedule the import task to run after the exclusionary period.

`--reschedule-offset=<offset-spec>`	Specify the amount of time before or after the exclusionary period to perform a user import. Use the following syntax: `--reschedule-offset="<number-of-units> <unit-type>"` `<number-of-units>` An integer from 1 to 2147483648 `<unit-type>` Seconds, minutes, hours, days, weeks, or months

Example

This example add a schedule in which users will be imported automatically into the MyNIS authentication service every 2 hours. In this example, the time zone is GMT-05:00 New York.

```
$ avaki nis --add-schedule MyNIS --periodic-schedule=
"2 hours" --time-zone="GMT-05:00 New York"
```

avaki nis --delete

Delete an NIS authentication service or delete users from an NIS authentication service.

Syntax

To delete an authentication service:

```
avaki nis {--delete | -d} <auth-service>
```

To delete users:

```
avaki nis {--delete | -d} {--user | -u}
  {<user-name>+ | {--all | -a}} <auth-service>
```

`<auth-service>`	The name of the authentication service to delete.
`--user | -u`	Delete one or more users from the specified authentication service.

`<user-name>+`	The name or names of user accounts to delete. To specify more than one user, enter a space-delimited list of user IDs.
`--all | -a`	Delete all users from the specified authentication service.

avaki nis --delete-schedule

Delete a schedule that specifies when users will be imported automatically.

To use this command, you need write permission on the NIS authentication service.

Syntax

```
avaki nis --delete-schedule <auth-service> <schedule-id>
```

`<auth-service>`	The name of the authentication service for which the schedule will be deleted.
`<schedule-id>`	The numeric ID of the schedule to delete. To determine a schedule's ID, use **avaki nis --list-schedules** (page 185).

avaki nis --import

Import user accounts or groups from an NIS authentication service into the current Avaki domain.

Note When you import a user, you import all the groups to which the user belongs. When you import a group, you import all users in the group and all groups to which those users belong.

Syntax

To import user accounts:

```
avaki nis {--import | -i} {--user | -u} {<user-name>+ | [--all
   | -a]} <auth-service>
```

To import groups:

```
avaki nis {--import | -i} {--group | -g} {<group-name>+ |
   [--all | -a]} <auth-service>
```

`--user | -u`	Import user accounts from the specified authentication service.
`{<user-name>+ | [--all | -a]}`	The name or names of the user accounts to import. To import all users, specify --all or -a. To specify one or more users, enter a space-delimited list of users.
`--group | -g`	Import groups from the specified authentication service.
`{<groups> | [--all | -a]}`	The name or names of the groups to import. To import all groups, specify --all or -a. To specify one or more groups, enter a space-delimited list of groups.
`<auth-service>`	The name of the authentication service to import from.

avaki nis --info

Display information about an NIS authentication service.

Syntax

```
avaki nis --info <auth-service>
```

`<auth-service>`	Specify the name of the service.

avaki nis --integrate

Integrate an NIS authentication service.

Syntax

```
avaki nis --integrate <auth-service> --nis-server=<nis-server>
  --nis-domain=<nis-domain> [--refresh-on-login]
```

`<auth-service>`	Devise a name for the new service.
`--nis-server=<nis-server>`	The name or IP address of the host machine on which the NIS authentication service resides.
`--nis-domain=<nis-domain>`	The name of the NIS domain for the authentication service you want to integrate.
`--refresh-on-login`	Optional. Refresh a user's information (such as group memberships) from the external directory each time the user logs in.

avaki nis --list-schedules

Display a list of schedules that specify when users will be imported automatically.

Syntax

```
avaki nis --list-schedules <auth-service>
```

`<auth-service>`	The name of the authentication service for which the schedule will be deleted.

avaki nis --update

Update an NIS authentication service.

Syntax

```
avaki nis --update <auth-service> --nis-server=<nis-server>
  --nis-domain=<nis-domain>
```

`<auth-service>`	Specify the name of the service.
`--nis-server=<nis-server>`	The name or IP address of the host machine on which the NIS authentication service resides.
`--nis-domain=<nis-domain>`	The name of the NIS domain for the authentication service you want to modify.

avaki passwd

Change the password for an Avaki user account.

Note Do not use this command on an account imported from an external directory service.

Syntax

```
avaki passwd [--user=<user>] {--old-password | --old}=<old>
  {--new-password | --new}=<new> [--force | -f]
```

`--user=<user>`	Optional. An Avaki account name. If you do not enter an account name on the command line, the system assumes you are changing your own password. To specify a user other than yourself, you must be a member of the Administrators group, and you must use the --force option.
`{--old-password | --old}=<old>`	The current password. If you're executing this command on a Unix system and your password contains non-alphanumeric characters (! or &, for instance), enclose your password in single quotation marks to prevent the shell from interpreting the special characters.
`{--new-password | --new}=<new>}`	The new password. If you're executing this command on a Unix system and your password contains non-alphanumeric characters (! or &, for instance), enclose your password in single quotation marks to prevent the shell from interpreting the special characters.
`--force | -f`	(Administrators only) Do not display the prompt to enter the old password.

avaki permissions

Display read, write, execute, and delete permissions for each user or group in the access control list for an object.

Syntax

```
avaki permissions <grid-path>+
```

`<grid-path>+`	The path to the grid file or grid directory for which you are viewing permissions. You can enter an absolute path or a relative one.

avaki plugin --generate

Run the data service Plug-in Wizard to generate skeleton Java code and build.xml and manifest files for a data service plug-in. For more information about writing your own data service plug-ins, see the *Sybase Avaki EII Provisioning and Advanced Data Integration Guide*.

Syntax

```
avaki plugin --generate [--plugin-name=<plugin-name>]
   [--package=<package-name>]
   [--impl-class=<implementation-class-name>]
   [--method-name=<implementation-method>]
   [--parameter=<parameter-spec>]* [--input=<stream-spec>]*
   [--output=<stream-spec>]*
   [--target-dir=<output-target-path>]
   [--api-classes=<api-class-location>]
   [--template-dir=<template-path>]
```

`--plugin-name=<plugin-name>`	Optional. The name of the plug-in. If this option is not specified, the default value is MyPlugin. The plug-in name will be included in the manifest file. The JAR file will be named <plugin-name>.jar.

`--package=<package-name>`	Optional. The package name of the implementation class. For example, if you specify the value “com.avaki.mypackage” the plug-in implementation class will be placed in the specified package. If this option is not specified, the implementation class will not be put in any packages.
`--impl-class=<implementation-class-name>`	Optional. The class name of the plug-in implementation that the tool will generate. If this option is not specified, the class name is the value of <plugin-name>.
`--method-name=<implementation-method>>`	Optional. The name of the method that implements the plug-in. The plug-in tool creates an empty method, and a developer adds custom code to it. The default value is plugin.
`--parameter=<parameter-spec>*`	Optional. A parameter specification for this plug-in. If the plug-in accepts more than one parameter, you can specify multiple --parameter options. Use the following form to specify a parameter: <name>=<value> Enclose parameters in double quotation marks (""). For example: `--parameter="name=myString;type=VARCHAR"` If you specify multiple parameters, use semicolons (;) to separate them. Specify any of the following names and values: name — A unique name for this parameter type — The type of this parameter, which can be one of the following:

ARRAY	INTEGER
BIGINT	JAVA_OBJECT
BINARY	LONGVARCHAR
BIT	NUMERIC
BOOLEAN	REAL
BLOB	SMALLINT
CHAR	TIME
CLOB	TIMESTAMP
DATE	TINYINT
DECIMAL	VARBINARY
DOUBLE	VARCHAR
FLOAT	

islist — Indicates whether this parameter's value is a list or a singleton. The possible values are true or false. The default is false.

The following example generates a plug-in that accepts two parameters:

```
avaki plugin -generate
--parameter="name=param1;type=INTEGER"
--parameter="name=param2;type=
VARCHAR,islist=true"
```

In this example, the type for the second parameter, "param2," is a list of strings.

`--input=<stream-spec>` — Optional. The input specification for the plug-in.

Specify one or more equal-sign-separated <name>=<value> pairs, where <name> is a unique name for the input specification and <value> is its value. If you enter more than one <name>=<value> pair, use a semicolon (;) to separate the pairs.

Specify the following names and values:

name — A unique name for this input.

type — The stream type of this input data, which can be one of the following:

- XML: Custom data in XML format
- ByteStream: Bytes of raw data
- ResultSet: Database result rowsets

	islist — (Optional) Indicates whether this input is a list or a single item. The possible values are true or false. The default is false
`--output=<stream-spec>`	Optional. The output destination for the plug-in. Specify one or more equal sign-separated <name>=<value> pairs, where <name> is a unique name for the output specification and <value> is its value. If you enter more than one <name>=<value> pair, use a semicolon (;) to separate the pairs. Specify the following names and values: name — A unique name for this output type — The stream type of this output data, which can be one of the following: • XML: Custom data in XML format • ByteStream: Bytes of raw data • ResultSet: Database result rowsets islist — (Optional) Indicates whether this output is a list or a singleton. The possible values are true or false. The default is false.
`--target-dir=<output-target-path>`	Optional. The name of the directory in which the plug-in wizard will store the generated code. The default location is ./plugin_src. If the necessary directories do not exist, they will be created.
`--api-classes=<api-class-location>`	Optional. The location of the Avaki API classes that are required to compile the generated plug-in code. You can specify a JAR file or the path to the classes. The default location is <avaki-install-dir>/apilib/javaki_plugin_api.jar.
`--template-dir=<template-path>`	Optional. The location of the directory that contains template files the plug-in wizard reads when generating code. The default location is <avaki-install-dir>/templates/plugins.

Example

```
$ avaki plugin -generate -plugin-name=Example
--package=com.avaki.examples
--impl-class=MyExample --method-name=myRunMethod
--parameter="name=myInt;type=INTEGER"
--parameter="name=myString;type=VARCHAR"
--input="name=input1;type=XML"
--input="name=input2;type=XML"
--output="name=myOutput;type=XML"
```

avaki proxy --add

Add another Avaki domain's proxy server to your Avaki domain's proxy routing table. This is part of the procedure for establishing an interconnection from one domain to another.

Note If anyone modifies your domain's proxy routing table during a CLI session, execute the command **avaki client --connect** (page 46) to force your command client to pick up the new routing information.

Syntax

```
avaki proxy {--add | -a} [--unencrypted] [--compress-data]
  <site-ID> {<proxy-server> <proxy-port> |
  <proxy-server>:<proxy-port>}
```

`--unencrypted`	Optional. Turn off SSL encryption, which is enabled by default. When SSL is disabled, performance will improve, but data is not secure. SSL encryption should not be disabled unless both grid domains are within a larger secure environment, such as behind a corporate firewall.
`--compress-data`	Optional. Use data compression when communicating with this proxy server.
`<site-ID>`	The name of the Avaki domain to connect with.

`<proxy-server>`	The IP address or DNS name of the machine on which the other domain's proxy server is running. We recommend that you enter the proxy host's IP address rather than its DNS name.
`<proxy-port>`	The proxy server's HTTP or HTTPS listening port. These are commonly: HTTP: 18080 HTTPS: 18443

avaki proxy --delete

Delete a proxy server from your Avaki domain's proxy routing table.

Syntax

```
avaki proxy {--delete | -d} <site-ID> {<proxy-server>
  <proxy-port> | <proxy-server>:<proxy-port>}
```

`<site-ID>`	The name of the Avaki domain to which the proxy server belongs.
`<proxy-server>`	The IP address or DNS name of the proxy server to be deleted.
`<proxy-port>`	The proxy server's HTTP or HTTPS listening port. These are commonly: HTTP: 18080 HTTPS: 18443

avaki proxy --list

List the proxy server entries in the proxy routing table.

Syntax

```
avaki proxy {--list | -l}
```

– This command has no options.

avaki pwd

Display the name of the current Avaki directory. Similar to the Unix **pwd** command.

Syntax

```
avaki pwd
```

– This command has no options.

avaki replica --add

Add a secondary Avaki domain controller to a domain.

Syntax

```
avaki replica {--add | -a} <machine>[:<port>]
```

`<machine>[:<port>]`	Specify the IP address or DNS name of the machine on which the secondary GDC is running. If the secondary GDC is using a connect port other than 3099, the default, specify the connect port number.

avaki replica --config

Configure the refresh interval for a secondary GDC.

Syntax

```
avaki replica {--config | -c}
  {--interval}=<backup-interval-in-seconds>
```

`--interval=<backup-interval-in-seconds>`	How frequently (in seconds) the secondary GDC is backed up. We recommend a minimum of 900 seconds (15 minutes).

avaki replica --delete

Delete a secondary grid domain controller from an Avaki domain.

Syntax

```
avaki replica {--delete | -d} <machine>[:<port>]
```

`<machine>[:<port>]`	The IP address or DNS name of the machine on which the secondary GDC is running. If the secondary GDC is using a connect port other than 3099, the default, specify the connect port number.

avaki replica --info

View the names and settings of any secondary GDCs in your grid domain.

Syntax

```
avaki replica --info
```

– This command has no options.

avaki replica --synch

Force an immediate update of a secondary GDC.

Syntax

```
avaki replica --synch
```

– This command has no options.

avaki rm

Delete a file, an Avaki directory, or the contents of an Avaki share.

Caution When you delete a shared file or directory from a grid domain, the change is reflected in the source file system—not in the grid only. If you want to shut down a share without affecting the source files, use **avaki server --share --stop** on page 231).

Syntax

```
avaki rm [--recursive | -r] [--force | -f] <grid-path>
```

`--recursive	-r`	Optional. If you use -r to delete a directory, the system deletes all the directory's contents, including all subdirectories and their contents.
`--force	-f`	Optional. Force the file, grid directory, or contents of a share to be removed from the grid directory.
`<grid-path>`	The path to the file, directory, or share to be deleted. The file, directory, or share must be in a grid directory. You can enter an absolute path or a relative one.

avaki schedule --delete

Delete a schedule from the specified server.

Syntax

```
avaki schedule --delete <server-name> <schedule-id>
```

`<server-name>` The name of the Avaki server from which the schedule will be deleted. For example, Bedrock.sybase.com.

`<schedule-id>` The numeric ID of the schedule to delete. To determine a schedule's ID, use **avaki schedule --list** (page 198).

avaki schedule --info

Display information about the specified schedule.

Syntax

```
avaki schedule --info <server-name> <schedule-id>
```

`<server-name>` The name of the Avaki server for which information will be displayed. For example, Bedrock.sybase.com.

`<schedule-id>` The numeric ID of the schedule. To determine a schedule's ID, use **avaki schedule --list** (page 198).

avaki schedule --list

List the schedules registered on the specified server.

Syntax

```
avaki schedule --list <server-name>
```

`<server-name>`	The name of the Avaki server for which schedules will be listed. For example, Bedrock.sybase.com.

avaki schedule --print-iterations

List the times when the specified schedule will execute.

Syntax

```
avaki schedule --print-iterations <server-name> <schedule-id>
  <num-iterations>
```

`<server-name>`	The name of the Avaki server for the execution times will be listed. For example, Bedrock.sybase.com.
`<schedule-id>`	The numeric ID of the schedule. To determine a schedule's ID, use **avaki schedule --list** (page 198).
`<num-iterations>`	The number of execution times to display.

avaki scheduleexclusion --create --custom

Create a custom exclusion to a schedule.

Syntax

```
avaki scheduleexclusion --create --custom
  [--description=<description-text>] [--time-zone=<time-zone>]
  <schedule-exclusion-name>
  {<start-time-stamp> <end-time-stamp>}+
```

`--description=<description-text>`

Specify a description for the exclusion.

`--time-zone=<time-zone>`

Specify whether the time zone for the excluded period is the grid server's local time or relative to Greenwich Mean Time (GMT). The format for the time zone is as follows:

GMT+12:00 Fiji	GMT+02:00 Cairo
GMT+11:00 Noumea	GMT+01:00 Paris
GMT+10:00 Sydney	GMT London
GMT+9:30 Adelaide	GMT-01:00 Azores
GMT+09:00 Tokyo	GMT-02:00 Mid-Atlantic
GMT+08:00 Hong Kong	GMT-03:00 Rio de Janeiro
GMT+07:00 Bangkok	GMT-03:30 Newfoundland
GMT+06:30 Rangoon	GMT-04:00 Caracas
GMT+06:00 Dacca	GMT-05:00 New York
GMT+05:45 Katmandu	GMT-06:00 Chicago
GMT+05:30 Calcutta	GMT-07:00 Denver
GMT+05:00 Karachi	GMT-08:00 San Francisco
GMT+04:30 Kabul	GMT-09:00 Juneau
GMT+04:00 Baku	GMT-10:00 Hawaii

	GMT+03:30 Tehran GMT-11:00 Samoa GMT+03:00 Moscow
`<schedule-exclusion-name>`	Specify a name for the schedule exclusion.
`{<start-time-stamp> <end-time-stamp>}+`	Specify one or more ranges of times and dates during which the exclusion will be in effect, in this format: `{"<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy>" "<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy>"}`

avaki scheduleexclusion --create --daily

Create a daily exclusion to a schedule.

Syntax

```
avaki scheduleexclusion --create --daily
  [--description=<description-text>] [--time-zone=<time-zone>]
  <schedule-exclusion-name> <start-time-of-day>
  <end-time-of-day>
```

`--description=<description-text>` — Specify a description for the exclusion.

`--time-zone=<time-zone>` — Specify whether the time zone for the excluded period is the grid server's local time or relative to Greenwich Mean Time (GMT). The format for the time zone is as follows:

GMT+12:00 Fiji	GMT+02:00 Cairo
GMT+11:00 Noumea	GMT+01:00 Paris
GMT+10:00 Sydney	GMT London
GMT+9:30 Adelaide	GMT-01:00 Azores
GMT+09:00 Tokyo	GMT-02:00 Mid-Atlantic
GMT+08:00 Hong Kong	GMT-03:00 Rio de Janeiro
GMT+07:00 Bangkok	GMT-03:30 Newfoundland
GMT+06:30 Rangoon	GMT-04:00 Caracas
GMT+06:00 Dacca	GMT-05:00 New York
GMT+05:45 Katmandu	GMT-06:00 Chicago
GMT+05:30 Calcutta	GMT-07:00 Denver
GMT+05:00 Karachi	GMT-08:00 San Francisco
GMT+04:30 Kabul	GMT-09:00 Juneau
GMT+04:00 Baku	GMT-10:00 Hawaii

	GMT+03:30 Tehran GMT-11:00 Samoa GMT+03:00 Moscow
`<schedule-exclusion-name>`	Specify a name for the schedule exclusion.
`<start-time-of-day>` `<end-time-of-day>`	Specify the range of times and dates during which the exclusion will be in effect, in this format: `"<hh>:<mm>:<ss> {am | pm}" "<hh>:<mm>:<ss> {am | pm}"`

avaki scheduleexclusion --create --monthly

Create a monthly exclusion to a schedule.

Syntax

```
avaki scheduleexclusion --create --monthly
  [--description=<description-text>] [--time-zone=<time-zone>]
  {--month-day=<month-day>}+ <schedule-exclusion-name>
  [<start-time-of-day> <end-time-of-day>]
```

`--description=<description-text>` — Specify a description for the exclusion.

`--time-zone=<time-zone>` — Specify whether the time zone for the excluded period is the grid server's local time or relative to Greenwich Mean Time (GMT). The format for the time zone is as follows:

GMT+12:00 Fiji	GMT+02:00 Cairo
GMT+11:00 Noumea	GMT+01:00 Paris
GMT+10:00 Sydney	GMT London
GMT+9:30 Adelaide	GMT-01:00 Azores
GMT+09:00 Tokyo	GMT-02:00 Mid-Atlantic
GMT+08:00 Hong Kong	GMT-03:00 Rio de Janeiro
GMT+07:00 Bangkok	GMT-03:30 Newfoundland
GMT+06:30 Rangoon	GMT-04:00 Caracas
GMT+06:00 Dacca	GMT-05:00 New York
GMT+05:45 Katmandu	GMT-06:00 Chicago
GMT+05:30 Calcutta	GMT-07:00 Denver
GMT+05:00 Karachi	GMT-08:00 San Francisco
GMT+04:30 Kabul	GMT-09:00 Juneau
GMT+04:00 Baku	GMT-10:00 Hawaii

	GMT+03:30 Tehran GMT-11:00 Samoa GMT+03:00 Moscow
{--month-day=<month-day>}+	Specify the days of the month during which the exclusion will be in effect, in this format: { 1 - 31 | L } L represents the last day of the month.
<schedule-exclusion-name>	Specify a name for the schedule exclusion.
[<start-time-of-day> <end-time-of-day>]	Optional. Specify the range of times and dates during which the exclusion will be in effect, in this format: "<hh>:<mm>:<ss> {am | pm}" "<hh>:<mm>:<ss> {am | pm}"

avaki scheduleexclusion --create --weekly

Create a weekly exclusion to a schedule.

Syntax

```
avaki scheduleexclusion --create --weekly
  [--description=<description-text>] [--time-zone=<time-zone>]
  {--weekday=<weekday-name>}+ <schedule-exclusion-name>
  [<start-time-of-day> <end-time-of-day>]
```

`--description=<description-text>` — Specify a description for the exclusion.

`--time-zone=<time-zone>` — Specify whether the time zone for the excluded period is the grid server's local time or relative to Greenwich Mean Time (GMT). The format for the time zone is as follows:

GMT+12:00 Fiji	GMT+02:00 Cairo
GMT+11:00 Noumea	GMT+01:00 Paris
GMT+10:00 Sydney	GMT London
GMT+9:30 Adelaide	GMT-01:00 Azores
GMT+09:00 Tokyo	GMT-02:00 Mid-Atlantic
GMT+08:00 Hong Kong	GMT-03:00 Rio de Janeiro
GMT+07:00 Bangkok	GMT-03:30 Newfoundland
GMT+06:30 Rangoon	GMT-04:00 Caracas
GMT+06:00 Dacca	GMT-05:00 New York
GMT+05:45 Katmandu	GMT-06:00 Chicago
GMT+05:30 Calcutta	GMT-07:00 Denver
GMT+05:00 Karachi	GMT-08:00 San Francisco
GMT+04:30 Kabul	GMT-09:00 Juneau
GMT+04:00 Baku	GMT-10:00 Hawaii

	GMT+03:30 Tehran GMT-11:00 Samoa GMT+03:00 Moscow
`{--weekday= <weekday-name>}+`	Specify the days of the week during which the exclusion will be in effect, in this format (the format is not case-sensitive): `{Sun | Mon | Tue | Wed | Thu | Fri | Sat}`
`<schedule-exclusion-name>`	Specify a name for the schedule exclusion.
`[<start-time-of-day> <end-time-of-day>]`	Optional. Specify the range of times and dates during which the exclusion will be in effect, in this format: `"<hh>:<mm>:<ss> {am | pm}" "<hh>:<mm>:<ss> {am | pm}"`

avaki scheduleexclusion --create --yearly

Create a yearly exclusion to a schedule.

Syntax

```
avaki scheduleexclusion --create --yearly
  [--description=<description-text>] [--time-zone=<time-zone>]
  {--date=<mm>-<dd>}+ <schedule-exclusion-name>
  [<start-time-of-day> <end-time-of-day>]
```

`--description=<description-text>` — Specify a description for the exclusion.

`--time-zone=<time-zone>` — Specify whether the time zone for the excluded period is the grid server's local time or relative to Greenwich Mean Time (GMT). The format for the time zone is as follows:

GMT+12:00 Fiji	GMT+02:00 Cairo
GMT+11:00 Noumea	GMT+01:00 Paris
GMT+10:00 Sydney	GMT London
GMT+9:30 Adelaide	GMT-01:00 Azores
GMT+09:00 Tokyo	GMT-02:00 Mid-Atlantic
GMT+08:00 Hong Kong	GMT-03:00 Rio de Janeiro
GMT+07:00 Bangkok	GMT-03:30 Newfoundland
GMT+06:30 Rangoon	GMT-04:00 Caracas
GMT+06:00 Dacca	GMT-05:00 New York
GMT+05:45 Katmandu	GMT-06:00 Chicago
GMT+05:30 Calcutta	GMT-07:00 Denver
GMT+05:00 Karachi	GMT-08:00 San Francisco
GMT+04:30 Kabul	GMT-09:00 Juneau
GMT+04:00 Baku	GMT-10:00 Hawaii

	GMT+03:30 Tehran GMT-11:00 Samoa GMT+03:00 Moscow

`{--date=<mm>-<dd>}+`	Specify the days of the month during which the exclusion will be in effect, in this format:
	`<mm>-{<dd> | L}`
	L represents the last day of the month. For example, both of the following specify October 31:
	`--date=10-31`
	`--date=10-L`

`<schedule-exclusion-name>`	Specify a name for the schedule exclusion.
`[<start-time-of-day> <end-time-of-day>]`	Optional. Specify the range of times and dates during which the exclusion will be in effect, in this format: `"<hh>:<mm>:<ss> {am | pm}" "<hh>:<mm>:<ss> {am | pm}"`

avaki scheduleexclusion --delete

Delete a schedule exclusion from the grid domain.

Syntax

```
avaki scheduleexclusion --delete <schedule-exclusion-name>
```

`<schedule-exclusion-name>`	Specify the name of the schedule exclusion to delete.

avaki scheduleexclusion --info

Display information about a specified schedule exclusion.

Syntax

```
avaki scheduleexclusion --info <schedule-exclusion-name>
```

`<schedule-exclusion-name>`	Specify the name of the schedule exclusion about which information will be displayed.

avaki scheduleexclusion --list

List the names of all schedule exclusions in the grid domain.

Syntax

```
avaki scheduleexclusion --list
```

_	This command has no options.

avaki search --create

Set up a search service to index attributes for all objects in a particular grid directory.

Syntax

```
avaki search --create [--include=<attribute-name>]
  [--exclude=<attribute-name>] [--interval=<rehash-interval>]
  [{--server | -s}=<grid-server>] [--recursive | -r]
  <search-service-name> <grid-directory-to-index>
```

`--include=<attribute-name>`	Optional. Specify which attributes the search service will include in the index.
`--exclude=<attribute-name>`	Optional. Specify which attributes the search service will exclude from the index.
`--interval=<reindex-interval>`	Optional. How often (in seconds) attribute information will be updated. Use the --interval flag when creating a search service; to change the reindex interval for an existing search service, use the avaki rehash command.
`{--server | -s}=<grid-server>`	Optional. Specify the name of the Avaki server on which the search service will be created. For example, Bedrock. If you omit this option, the search service is created on the current server.
`[--recursive | -r]`	Optional. Recursively search the specified grid directory and all its contents.
`<search-service-name>`	Specify a name for the search service.
`<grid-directory-to-index>`	The root directory from which the attributes will be collected. All searches using a particular search service are recursive—that is, they search everything under the <grid-directory-to-index>.

avaki search --delete

Delete a search service from a grid domain.

Syntax

```
avaki search {--delete | -d} <search-service-name>
  [{--domain | -D}=<domain>]
```

`<search-service-name>`	The name of the search service you're deleting.
`{--domain | -D}=<domain>`	Optional. The grid domain of the search service to delete. If you do not specify a domain, the system assumes the path is in the current grid domain.

avaki search (execute)

Perform a search for grid objects (such as files and directories) that have a particular attribute or attribute value. To use this command, you must previously have created a search service (see **avaki search --create** on page 210).

Note The web UI has more powerful search capabilities than the CLI, including the ability to search for values that are greater than, less than, or not equal to the specified <search-value>. See the *Sybase Avaki EII Administration Guide* for instructions on using the search tools in the web UI.

Syntax

```
avaki search --name=<attribute-name> [--value=<search-value>]
   [{--operation | -op}=<operation>] [{--type | -t}=<type>]
   <search-service-name> [{--domain | -D}=<domain>]
```

`--name=<attribute-name>`

Specify the attribute to search for. For example, to search for a file by name, enter --name=system/name. (You can use the command **avaki attribute --list** (page 21) to list other attributes to search for.)

`--value=<search-value>`

Optional. Specify a value for the attribute in the --name option. If you use this option, the search returns a list of grid objects (files, for example) for which the target attribute is set to <search-value>. If you omit this option, the search returns a list of all grid objects that have the target attribute.

You can use a wildcard to indicate a character or set of characters to use in searches. A wildcard can indicate a point in a search string where any character or string is a good match, where any of one or more specified characters is a good match, or where any of the specified characters is not an acceptable match. The following wildcard characters may be used in searches:

% or *	Match any character or string at this position in the search pattern
_ or ?	Match any single character at this position in the search pattern

`{--operation | -op}=<operation>`

Optional. Specify a logical operator to be used in the search, in this format:

=	is equal to
<>	is not equal to
>	is greater than
>=	is greater than or equal to
<	is less than
<=	is less than or equal to

If --operation is not specified, <operation> is assumed to be = (is equal to).

`{--type | -t}=<type>`

Specify the type of the attribute to search for.

string

integer

real

date

time

timestamp

If --type is not specified, <type> is assumed to be string.

`<search-service-name>`	Specify the name of the search service to use for this search.
`{--domain | -D}=<domain>`	Optional. The grid domain of the search service to use. If you do not specify a domain, the system assumes the path is in the current grid domain.

avaki search --get-rehash-interval

View the rehash interval for a search service.

Syntax

```
avaki search --get-rehash-interval <search-service-name>
  [{--domain | -D}=<domain>]
```

`<search-service-name>`	The name of the search service whose rehash interval you're viewing.
`{--domain | -D}=<domain>`	Optional. The grid domain of the search service whose rehash interval you're viewing. If you do not specify a domain, the system assumes the path is in the current grid domain.

avaki search --info

Display information about a search service.

Syntax

```
avaki search {--info | -i} <search-service-name>
  [{--domain | -D}=<domain>]
```

`<search-service-name>`	The name of the search service whose information you want to display.
`{--domain | -D}=<domain>`	Optional. The grid domain of the search service to display information about. If you do not specify a domain, the system assumes the path is in the current grid domain.

avaki search --rehash

Forces a search service to reindex its contents.

Syntax

```
avaki search --rehash <search-service-name>
  [{--domain | -D}=<domain>]
```

`<search-service-name>`	The name of the search service you're rehashing.
`{--domain | -D}=<domain>`	Optional. The grid domain of the search service to rehash. If you do not specify a domain, the system assumes the path is in the current grid domain.

avaki search --set-rehash-interval

Configure the rehash interval for a search service.

Syntax

```
avaki search --set-rehash-interval <search-service-name>
  [{--domain | -D}=<domain>] <new-rehash-interval>
```

`<search-service-name>`	The name of the search service whose rehash interval you're configuring.
`{--domain | -D}=<domain>`	Optional. The grid domain of the search service whose rehash interval you're configuring. If you do not specify a domain, the system assumes the path is in the current grid domain.
`<new-rehash-interval>`	Specify how frequently (in seconds) updates are propagated from the local file system into the shared directory in the grid. Default: 0 (never rehash) Range: 0 to 2147483648 seconds (about 68 years)

avaki security --config

Configure the encryption level for all files in an Avaki directory or Avaki share.

To display the encryption level for a grid object, use **avaki security --info** (page 222).

Syntax

```
avaki security --config --encryption-level={clear | encrypted}
  <target-grid-path>
```

`--encryption-level={clear | encrypted}`	Specify the encryption level for the files: clear (encryption is turned off for this directory or share) or encrypted (encryption is turned on).
`<target-grid-path>`	The path to the file or directory for which you are modifying the encryption level.

avaki security --default-gid

Configure the default local GID for groups in a specified grid authentication service, or display the current default local GID for the specified authentication service. When no specific mapping exists for a group in that authentication service, the data grid access server allows members of the group to access grid data (subject to grid access controls) and passes the specified GID to the NFS client when the client needs to know the ownership of a file or directory. If you do not configure a default GID, users in the target authentication service who are not covered by specific user or group mappings will not be able to access grid data through the DGAS.

Syntax

To display the current default GID:

```
avaki security --default-gid
```

To configure a default GID:

```
avaki security --default-gid [--auth-service-path=<path>]
  [<gid>]
```

`--auth-service-path=<path>`	Optional. The parent directory of the grid authentication service in which to configure the default GID. For example, /System/Domains/Bedrock/Services/AuthServices/Ldap/BedrockLdap. The default is /System/Domains/<domain-name>/Services/AuthServices/Grid/DefaultAuthService.
`<gid>`	Optional. Specify the default GID. If you omit this option, the system displays the current default local GID.

avaki security --default-group

Use this command to configure a DGAS to use the specified group name as the default group from within the specified authentication service path if an unknown or unmapped user accesses the grid from an NFS client.

Syntax

```
avaki security --default-group [--auth-service-path=<path>]
  [<group-name>]
```

`--auth-service-path=<path>`	Optional. The parent directory of the specified or default group's grid authentication service. For example, /System/Domains/Bedrock/Services/AuthServices/Ldap/BedrockLdap. The default is /System/Domains/<domain-name>/Services/AuthServices/Grid/DefaultAuthService.
`<group-name>`	Optional. The name of the group in the specified authentication service that you are configuring as the default grid group for client groups to map to. If you omit this option, the system displays the default group name.

avaki security --default-uid

Configure the default local UID for users in a specified grid authentication service, or display the current default local UID for the specified grid authentication service. When no specific mapping exists for a user in that authentication service, the data grid access server allows users to access grid data (subject to grid access controls) and passes the specified UID to the NFS client when the client needs to know the ownership of a file or directory. If you do not configure a default UID, users in the target authentication service who are not covered by specific user or group mappings will not be able to access grid data through the DGAS.

Syntax

To display the current default UID:

```
avaki security --default-uid
```

To configure a default UID:

```
avaki security --default-uid [--auth-service-path=<path>]
  [<uid>]
```

`--auth-service-path=<path>`	Optional. The parent directory of the grid authentication service for which you are configuring a default UID. For example, /System/Domains/Bedrock/Services/AuthServices/Ldap/BedrockLdap. The default is /System/Domains/<domain-name>/Services/AuthServices/Grid/DefaultAuthService.
`<uid>`	Optional. Specify the default UID. If you omit this option, the system displays the default UID.

avaki security --default-user

If you want to allow unknown or unmapped users to access the grid from NFS clients, configure a DGAS to use the specified user name as the default user from within the specified authentication service path.

Syntax

```
avaki security --default-user [--auth-service-path=<path>]
  [<user-name>]
```

`--auth-service-path=<path>`	Optional. The parent directory of the default user's grid authentication service. For example, /System/Domains/Bedrock/Services/AuthServices/Ldap/BedrockLdap. The default is /System/Domains/<domain-name>/Services/AuthServices/Grid/DefaultAuthService.
`<user-name>`	Optional. The name of the user in the specified authentication service that you are configuring as the default grid user for client users to map to. If you omit this option, the system displays the default user name.

avaki security --gid

Configure the NFS client group ID for the specified grid group, or display the current GID for the specified grid group.

Syntax

To display the current GID:

```
avaki security --gid --group=<path>
```

To configure a GID:

```
avaki security --gid --group=<path> <new-gid>
```

To remove a GID:

```
avaki security --gid --group=<path> --delete
```

`--group=<path>`	The grid directory of the group.
`--delete`	Optional. Remove the GID configured for the specified grid group.
`<new-gid>`	Optional. Specify the GID that you are mapping to the grid group.

avaki security --info

Display the encryption level for a grid directory, file, or share.

To set encryption levels, use **avaki security --config** (page 217).

Syntax

```
avaki security --info <target-grid-path>
```

`<target-grid-path>`	The path to the grid directory, file, or share for which you want to view the encryption level.

avaki security --uid

Configure the NFS client user ID for the specified grid user, or display the current UID for the specified grid user.

Syntax

To display the current UID:

```
avaki security --uid --user=<path>
```

To configure a UID:

```
avaki security --uid --user=<path> <new-uid>
```

To remove a UID:

```
avaki security --uid --user=<path> --delete
```

`--user=<path>`	The grid directory of the user. For example, /System/LocalDomain/Services/AuthServices/Grid/DefaultAuthService/Users/wilma.
`--delete`	Optional. Remove the UID configured for the specified grid user.
`<new-uid>`	Optional. Specify the UID that you are mapping to the grid user.

avaki server --dgas --connect

Connect a data grid access server to a grid domain. To use this command, you must be logged into Avaki as a member of the Administrators group.

These are the tasks you must complete to set up a DGAS:

1. If the DGAS will run on a Windows machine and if you want the DGAS to serve CIFS shares, disable native CIFS Windows services running on ports 445, 137, 138, and 139. See the *Sybase Avaki EII Administration Guide* for details.
2. (Optional.) Create a properties file for the DGAS. (See "Appendix B, "Setting up DGAS properties files" for more information.)
3. Use **dgas --start** to start the DGAS. See the *Sybase Avaki EII Administration Guide* for details.
4. Use **avaki server --dgas --connect** (page 224) to connect the DGAS to a grid domain.
5. Use **avaki dgas --set-property** (page 114) to set properties for the DGAS. (You can skip this step if you created a properties file in item 2.)
6. Use **avaki dgas --initialize** (page 105) to initialize the DGAS's credentials. If you created a properties file in item 2., specify the location of the properties file in this command.
7. Set up CIFS shares (**avaki dgas --create-cifs-share** (page 97)).
8. Access data catalog files on CIFS or NFS clients.

Syntax

```
avaki server --dgas --connect <machine>:[<port>] <dgas-name>
```

`<machine>:[<port>]`	The IP address or DNS name of the machine hosting the DGAS that you are connecting to the grid domain. If you want the DGAS to use a connect port other than 1399, the default, specify the connect port number.
`<dgas-name>`	The name you are assigning to the DGAS.

avaki server --dgas --destroy

Destroy a data grid access server and delete its state information and caches. To use this command, you must be logged in to Avaki as a member of the Administrators group.

Caution This command irrevocably destroys the DGAS. To halt the DGAS without destroying it or losing its state and caches, use **avaki server --dgas --stop**, below.

Syntax

```
avaki server --dgas --destroy [-f | --force] <dgas-name>
```

`-f | --force`	Optional. If you specify the -f option, the server will be removed from the grid even if the server is not running. If the server is running, any of the server's links to domainwide database operations (such as /System/LocalDomain/Services/DatabaseServices/MyDBConnector/MyDBOperation) or data services (such as /System/LocalDomain/Services/DataServices/MyDataService) will be removed.
`<dgas-name>`	The name of the DGAS to be destroyed.

avaki server --dgas --stop

Stop a data grid access server, but preserve its state information and caches. This command can be issued from a remote machine. (Use **dgas --start** (page 7) to restart a stopped DGAS.)

Syntax

```
avaki server --dgas --stop <dgas-name>
```

`<dgas-name>`	The name of the DGAS to be stopped.

avaki server --grid --connect

Connect a grid server to an Avaki domain.

Syntax

```
avaki server --grid --connect <machine>:[<port>]
```

`--connect <machine> :[<port]`	For <machine>, specify the IP address or DNS name of the machine hosting the Avaki server that you are connecting to the grid domain. For <port>, specify connect port number for the machine if the server is using a nondefault connect port.

avaki server --grid --destroy

Disconnect a grid server from an Avaki domain and destroy it.

Caution The **avaki server --grid --destroy** command irrevocably destroys the grid server. To halt the grid server without destroying it or losing its state and caches, use **avaki server --grid --stop** (page 228).

Syntax

```
avaki server --grid --destroy [-f | --force]
   [<machine>[:<port>]]
```

`--destroy [-f | --force] [<machine> [:<port>]]`

If you specify the -f option, the grid server will be removed from the Avaki domain even if the grid server is not running. If the server is running, any of the server's links to domainwide database operations (such as /System/LocalDomain/Services/DatabaseServices/MyDBConnector/MyDBOperation) or data services (such as /System/LocalDomain/Services/DataServices/MyDataService) will be removed.

For <machine>[:<port>], specify one of the following:

- If the grid server to be destroyed is alive, specify the IP address or DNS name of the machine hosting the Avaki server that you are destroying. If the server is using a nondefault connect port, specify the connect port number.
- If the grid server to be destroyed is not alive, specify the server's node name; do not specify a connect port. To determine the node name, execute the **avaki ls** command on the grid path to the Servers directory. For example, if you issue the following command:

  ```
  avaki ls /System/LocalDomain/Servers
  ```

  The result will be similar to the following:

  ```
  Servers (directory)
    . (directory)
    .. (directory)
    slatequarry.avaki.com (directory)
  ```

  In this example, the server's node name is: slatequarry.avaki.com

avaki server --grid --stop

Shut down an Avaki grid server. (To restart the grid server, see **grid-server --start** on page 11.)

Syntax

```
avaki server --grid --stop <machine>[:<port>]
```

`--stop <machine> [:<port>]`	The IP address or DNS name of the machine hosting the Avaki server that you are stopping. If the machine is on a nondefault connect port, specify the connect port number.

avaki server --proxy

Connect an Avaki firewall proxy server to a grid domain, destroy a firewall proxy server, or stop a firewall proxy server.

Caution The **avaki server --proxy --destroy** command irrevocably destroys the proxy server. To halt the proxy server without destroying it or losing its state and caches, use **avaki server --proxy --stop** (page 228).

Syntax

```
avaki server --proxy {--connect <machine>[:<port>] | --destroy
   | --stop} [-f | --force] <proxy-name>
```

`--connect <machine> [:<port>]`	Connect the Avaki proxy server on the specified machine to the current grid domain. For <machine>, enter the IP address or DNS name of the machine hosting the proxy server. If the proxy server is using a connect port other than 1199, the default, specify the connect port number.
`--destroy`	Delete the specified Avaki proxy server.

`--stop`	Stop (but do not destroy) a proxy server that is registered as a service. (To restart a proxy server, use **proxy-server --start** (page 14) or **proxy-server --register** (page 13).)
`-f | --force`	Optional. If you specify the -f option, the server will be removed from the grid even if the server is not alive. If the server is alive, any of the server's links to domainwide database operations (such as /System/LocalDomain/Services/DatabaseServices/MyDBConnector/MyDBOperation) or data services (such as /System/LocalDomain/Services/DataServices/MyDataService) will be removed.
`<proxy-name>`	The grid name of the proxy server.

avaki server --share --connect

Connects a share server to a grid server. You can connect a share server to multiple grid servers.

Syntax

```
avaki server --share --connect <share-server-logical-name>
  <share-server-host-name>[:<port>]
  <grid-server-host-name>[:<port>]
```

`<share-server-logical-name>`	The logical name of the new share server.
`<share-server-host-name>[:<port>]`	The machine name or IP address of the new share server. If you want the share server to use a connect port other than 2099, the default, specify the connect port number.
`<grid-server-host-name>[:<port>]`	The DNS name or IP address of the grid server to which to join this share server. If the grid server is using a connect port other than 3099, the default, specify the connect port number.

avaki server --share --disconnect

Disconnects a share server from a grid server. A share server may be connected to multiple grid servers.

Note Before you disconnect a share server, make sure there are no Avaki shares using the share server. You can disable the shares by taking them off line (see "avaki share --set-status" on page 244) or you can dissociate them from the share server (see "avaki share --remove-share-servers" on page 241)

Syntax

```
avaki server --share --disconnect [-f | --force]
  <share-server-name> <grid-server-name>
```

`-f | --force`	Optional. If you specify the -f option, the server will be removed from the grid even if the server is not alive. If the server is alive, any of the server's links to domainwide database operations (such as /System/LocalDomain/Services/DatabaseServices/MyDBConnector/MyDBOperation) or data services (such as /System/LocalDomain/Services/DataServices/MyDataService) will be removed.
`<share-server-name>`	The logical name of the share server.
`<grid-server-name>`	The DNS name or IP address of the grid server to disconnect from.

avaki server --share --stop

Shut down a share server. When you shut down a share server, the share server's shared directories and file names remain visible in the grid data catalog, but the shared files are inaccessible.

(You can restart the share server using the **share-server --start** (page 17) or **share-server --register** (page 16) command on the machine where the share server runs.)

Syntax

```
avaki server --share --stop <machine>[:<port>]
```

`<machine>[:<port>]`	The IP address or DNS name of the machine on which you are shutting down the share server. If the share server is using a starting port other than 2099, the default, specify the starting port number.

avaki share --add-rehash-schedule

Add a rehash (refresh) schedule for an Avaki share.

Syntax

```
avaki share --add-rehash-schedule <share-grid-path>
  [{--periodic-schedule=<period-spec> |
  --cron-schedule=<cron-spec> |
  --one-time-schedule=<time-stamp>}]
  [--start-time=<time-stamp>] [[--end-time=<time-stamp>] |
  [--max-iterations=<num-iterations>]]
  [--time-zone=<time-zone>] [--exclusion=<exclusion-name>]+
  [--exclusion-action={ drop | sooner | later }]
  [--reschedule-offset=<offset-spec>]
```

`<share-grid-path>`

The path of the root share directory in the data catalog. For example:

```
/Shares/MyShare
```

`--periodic-schedule=<period-spec>`

Specify a periodic schedule for running the rehash schedule. Use the following syntax:

```
--periodic-schedule="<number-of-units>
<unit-type>"
```

`<number-of-units>`	An integer from 1 to 2147483648
`<unit-type>`	Seconds, minutes, hours, days, weeks, or months

`--cron-schedule=<cron-spec>`

Specify how often a a cron-based scheduler for rehashing schedules will run the rehash schedule. Use the following syntax:

```
--cron-schedule="<seconds> <minutes> <hours>
<days-of-month> <months> <days-of-week> [<years>]"
```

For details about the <cron-spec> syntax, see Appendix C, “Configuring cron schedules”.

`--one-time-schedule=<time-stamp>`

Specify a single time and date for running the rehash schedule. Use the following syntax:

```
--one-time-schedule="{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> | now}"
```

`--start-time=<time-stamp>`

Specify the time and date that the rehash schedule will start running, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> | now}
```

`--end-time=<time-stamp>`

Specify the time and date that the rehash schedule will stop running, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> | now}
```

`--max-iterations=<num-iterations>`

Specify the number of times a schedule will repeat.

`<num-iterations>`	An integer from 1 to 2147483648. If no value is specified, the schedule repeats indefinitely.

`--time-zone=<time-zone>`

Specify the time zone for the schedule. The schedule can be specified according to the share server's time zone or relative to Greenwich Mean Time (GMT). If no value is specified for the time zone, the server's local time zone is used. The format for the time zone is as follows:

GMT+12:00 Fiji	GMT+02:00 Cairo
GMT+11:00 Noumea	GMT+01:00 Paris
GMT+10:00 Sydney	GMT London
GMT+9:30 Adelaide	GMT-01:00 Azores
GMT+09:00 Tokyo	GMT-02:00 Mid-Atlantic
GMT+08:00 Hong Kong	GMT-03:00 Rio de Janeiro
GMT+07:00 Bangkok	GMT-03:30 Newfoundland
GMT+06:30 Rangoon	GMT-04:00 Caracas
GMT+06:00 Dacca	GMT-05:00 New York
GMT+05:45 Katmandu	GMT-06:00 Chicago

	GMT+05:30 Calcutta	GMT-07:00 Denver
	GMT+05:00 Karachi	GMT-08:00 San Francisco
	GMT+04:30 Kabul	GMT-09:00 Juneau
	GMT+04:00 Baku	GMT-10:00 Hawaii
	GMT+03:30 Tehran	GMT-11:00 Samoa
	GMT+03:00 Moscow	

`--exclusion=<exclusion-name>`

The name of an exclusion to this schedule. For information about creating schedule exclusions, see **avaki scheduleexclusion --create --daily** (page 201), **avaki scheduleexclusion --create --weekly** (page 205), **avaki scheduleexclusion --create --monthly** (page 203), **avaki scheduleexclusion --create --yearly** (page 207), and **avaki scheduleexclusion --create --custom** (page 199).

`--exclusion-action={drop | sooner | later}`

Specify the action to take if the rehash schedule is scheduled to run during an exclusionary period.

drop	Don't run the rehash schedule during the exclusionary period.
sooner	Reschedule the rehash schedule to run before the exclusionary period.
later	Reschedule the rehash schedule to run after the exclusionary period.

`--reschedule-offset=<offset-spec>`

Specify the amount of time before or after the exclusionary period to rehash a schedule. Use the following syntax:

```
--reschedule-offset="<number-of-units>
<unit-type>"
```

`<number-of-units>`	An integer from 1 to 2147483648
`<unit-type>`	Seconds, minutes, hours, days, weeks, or months

avaki share --add-share-servers

Add one or more share servers to an Avaki share.

Syntax

```
avaki share --add-share-servers <share-grid-path>
  {--share-server=<share-server-name>[:<load-balance-factor>]}+
```

`<share-grid-path>`	The path of the root share directory in the grid. For example: `/Shares/MyShare`
`--share-server=<share-server-name>[:<load-balance-factor>]+`	Optional. The share servers for the share. The <share-server-name> name can be up to 32 characters long, and is specific to the grid server you are using. If you specify more than one name, you enable failover between the servers. For more information, see the *Sybase Avaki EII Administration Guide*. If you configure multiple share servers, the <load-balance-factor> specifies the approximate proportion of the load carried by <share-server-name>. The proportion is a fraction, as follows: • The numerator is the LBF, an integer that specifies the portion of load that a share server owns for a given Avaki share. • The denominator is the sum of the LBFs for all the share servers serving this Avaki share.

avaki share --create

Create an Avaki share (that is, share a local directory into the data catalog).

By default, a share does not reread the contents of the shared directory from the local file system. For information about adding a rehash (refresh) schedule to an Avaki share directory, see **avaki share --rehash** on page 241. For information about listing the rehash (refresh) events for an Avaki share, see **avaki share --list-rehash-schedules** on page 240. For information about deleting a rehash (refresh) schedule, see **avaki share --delete-rehash-schedule** on page 238.

Note When you create a share from a command client, Avaki looks for the <local-path> on the GDC machine if no share server or grid server is specified.

Syntax

```
avaki share --create <local-path> <share-grid-path>
   [--grid-server=<grid-server>]
   [--share-server=<share-server-name>[:<load-balance-factor>]+
   [--encryption-level={clear | encrypted}]
   [--background | --bg] [--owner=<initial-owner>]
```

`<local-path>`	The path to the directory in the local file system that you are sharing into the data catalog. This path must resolve to a valid name on all share servers associates with the shares. We recommend that you provide an absolute path name to avoid confusion. The <share-server> name can be up to 32 characters long.
`<share-grid-path>`	Specify the parent directory and subdirectory for the share. The <share-grid-path> can be up to 255 characters long. A grid directory with this name is placed inside the parent directory you have specified. This subdirectory contains the shared data.
`--grid-server=<grid-server>`	Optional. If you want the new share to use a particular grid server, specify the name of the grid server. For example, Bedrock.

`--share-server=<share-server-name>[:<load-balance-factor>]+`	Optional. The share servers for the share. The <share-server-name> name can be up to 32 characters long, and is specific to the grid server you are using. If you specify more than one name, you enable failover between the servers. For more information, see the *Sybase Avaki EII Administration Guide*. If you configure multiple share servers, the <load-balance-factor> specifies the approximate proportion of the load carried by <share-server-name>. The proportion is a fraction, as follows: • The numerator is the LBF, an integer that specifies the portion of load that a share server owns for a given Avaki share. • The denominator is the sum of the LBFs for all the share servers serving this Avaki share.
`--encryption-level={clear | encrypted}`	Optional. Specify the encryption level for the share: clear (encryption is turned off for this share) or encrypted (encryption is turned on).
`--background | --bg`	Optional. The share is be built in the background. The command line returns immediately.
`--owner=<initial-owner>`	Optional. Specify a share's owner. By default, new shares are owned by the user creating the share.

avaki share --delete-rehash-schedule

Delete a rehash (refresh) schedule from an Avaki share.

Syntax

```
avaki share --delete-rehash-schedule <share-grid-path>
  <schedule-id>
```

`<share-grid-path>`	The path to the shared directory in the local file system.
`<schedule-id>`	The numeric ID of the schedule to delete. To determine a schedule's ID, use **avaki share --list-rehash-schedules** (page 240).

avaki share --disconnect

Permanently disconnect and shut down an Avaki share and remove any links to that share.

Caution When you disconnect a shared directory from a grid domain, the change is not reflected in the source file system; any related source files remain on the source file system.

Syntax

```
avaki share --disconnect <share-grid-path>
```

`<share-grid-path>`	The path of the root share directory in the grid. For example: `/Shares/MyShare`

avaki share --get-local-path

Get the local path for an Avaki share.

Syntax

```
avaki share --get-local-path <share-grid-path>
```

`<share-grid-path>`	The path of the root share directory in the data catalog. For example: `/Shares/MyShare`

avaki share --get-status

Get the current status for an Avaki share: online or offline.

Syntax

```
avaki share --get-status <share-grid-path>
```

`<share-grid-path>`	The path of the root share directory in the data catalog. For example: `/Shares/MyShare`

avaki share --list-rehash-schedules

List the rehash (refresh) events for an Avaki share.

Syntax

```
avaki --list-rehash-schedules <share-grid-path>
```

`<share-grid-path>`	The path of the root share directory in the data catalog. For example: `/Shares/MyShare`

avaki share --list-share-servers

List the share servers for an Avaki share.

Syntax

```
avaki share --list-share-servers <share-grid-path>
```

`<share-grid-path>`	The path of the root share directory in the data catalog. For example: `/Shares/MyShare`

avaki share --rehash

Rehash (refresh) an Avaki share.

Syntax

```
avaki share --rehash <share-grid-path> [--background | --bg]
```

`<share-grid-path>`	The path of the root share directory in the data catalog. For example: `/Shares/MyShare`
`--background | --bg`	Optional. The share will be rehashed in the background. The command line returns immediately.

avaki share --remove-share-servers

Remove a share server from an Avaki share.

Syntax

```
avaki share --remove-share-servers <share-grid-path>
  {--share-server=<share-server-name>}+
```

`<share-grid-path>`	The path of the root share directory in the data catalog. For example: `/Shares/MyShare`
`--share-server=share-server-name`	The logical name of one or more share servers to remove.

avaki share --set-local-path

Set the local path for an Avaki share.

Syntax

```
avaki --set-local-path <share-grid-path> <local-path>
```

`<share-grid-path>`	The path of the root share directory in the data catalog. For example: `/Shares/MyShare`
`local-path`	The local path for the shared directory.

Example

```
$ avaki --set-local-path /Shares/MyAvakiShare c:\Bedrock
```

avaki share --set-share-servers

Replace the set of share servers for an Avaki share.

Syntax

```
avaki --set-share-servers <share-grid-path>
  {--share-server=<share-server-name>[:<load-balance-factor]}+
```

`<share-grid-path>`	The path of the root share directory in the data catalog. For example: `/Shares/MyShare`
`--share-server=<share-server-name>[:<load-balance-factor>]+`	The names of one or more share servers that will replace the set of share servers for <share-grid-path>. The <share-server> names can be up to 32 characters long. If you configure multiple share servers, the <load-balance-factor> specifies the approximate proportion of the load carried by <share-server-name>. The proportion is a fraction, as follows: • The numerator is the LBF, an integer that specifies the portion of load that a share server owns for a given Avaki share. • The denominator is the sum of the LBFs for all the share servers serving this Avaki share. For more information, see the *Sybase Avaki EII Administration Guide*.

avaki share --set-status

Setting the status for an Avaki share: online or offline.

Syntax

```
avaki --set-status <share-grid-path> {online | offline}
```

`<share-grid-path>`	The path of the root share directory in the grid. For example: `/Shares/MyShare`
`online | offline`	The new status for a shared directory: online or offline.

avaki share --update-share-servers

Modify the load balance factor for a set of share servers.

Syntax

```
avaki --update-share-servers <share-grid-path>
  {--share-server=<share-server-name>:<load-balance-factor>}+
```

`<share-grid-path>`	The path of the root share directory in the data catalog. For example: `/Shares/MyShare`
`--share-server=<share-server-name>[:<load-balance-factor>]+`	The names of one or more share servers that will replace the set of share servers for <share-grid-path>. The <share-server> names can be up to 32 characters long. If you configure multiple share servers, the <load-balance-factor> specifies the approximate proportion of the load carried by <share-server-name>. The proportion is a fraction, as follows: • The numerator is the LBF, an integer that specifies the portion of load that a share server owns for a given Avaki share. • The denominator is the sum of the LBFs for all the share servers serving this Avaki share. For more information, see the *Sybase Avaki EII Administration Guide*.

avaki shell

Open an Avaki shell. You can use an Avaki shell to issue troubleshooting commands or to work in an Avaki-only command environment. In the shell, do not enter the **avaki** part of Avaki commands. For example, for **avaki ls** you enter just **ls**.

Note To exit an Avaki shell, type **quit**.

Syntax

```
avaki shell [--debug | -d] [--perf | -p]
```

`--debug | -d` Optional. Toggles debug mode on/off. When debug mode is enabled, you can use it to troubleshoot problems.

`--perf | -p` Optional. Toggles performance tracking on/off. When performance tracking is enabled, you can view how long it takes to run a command that you issue within the shell.

avaki sqlview --delete

Delete a SQL view.

Syntax

```
sqlview --delete --type={dataservice | dbop | provisioned |
  virtual_dbop} <qualified-table-name>
```

`--type={dataservice | dbop | provisioned | virtual_dbop}` The name of the data service, database operation, provisioned table, or virtual database operation from which the SQL view will be deleted. The format for <qualified-table-name> is:

Data service	[<domain-name>.]DATASERVICE.<table-name>

Database operation	[<domain-name>.]<database-connector-name>.<table-name>
Provisioned table	[<domain-name>.]<database-connector-name>.<table-name>
Virtual database operation	[<domain-name>.]VIRTUALDB.<table-name>

avaki sqlview --get-description

Display a description for a SQL view.

Syntax

```
avaki sqlview --get-description --type={dataservice | dbop |
  provisioned | virtual_dbop} <qualified-table-name>
```

`--type={dataservice | dbop | provisioned | virtual_dbop}`

The name of the data service, database operation, provisioned table, or virtual database operation for which the SQL view's description will be displayed. The format for <qualified-table-name> is:

Data service	[<domain-name>.]DATASERVICE.<table-name>
Database operation	[<domain-name>.]<database-connector-name>.<table-name>
Provisioned table	[<domain-name>.]<database-connector-name>.<table-name>
Virtual database operation	[<domain-name>.]VIRTUALDB.<table-name>

avaki sqlview --set-description

Modify the description of a SQL view.

Syntax

```
avaki sqlview --set-description  --type={dataservice | dbop |
  provisioned | virtual_dbop} <qualified-table-name>
  <description>
```

`--type=`

`{dataservice | dbop | provisioned | virtual_dbop}`

The name of the data service, database operation, provisioned table, or virtual database operation for which the SQL view's description will be modified. The format for <qualified-table-name> is:

[<domain-name>.]<database-connector-name>.<database-operation-name>

Data service	[<domain-name>.]DATASERVICE.<table-name>
Database operation	[<domain-name>.]<database-connector-name>.<table-name>
Provisioned table	[<domain-name>.]<database-connector-name>.<table-name>
Virtual database operation	[<domain-name>.]VIRTUALDB.<table-name>

`<description>` Specify a description for the SQL view.

avaki status

Display the status of operations currently running on a grid server.

Syntax

```
avaki status --server=<grid-server> [--xml] [<opType>+]
```

`--grid-server=<grid-server>`	Optional. Specify the name of the grid server for which the status will be printed. For example, Bedrock.	
`--xml`	Optional. Print the output in XML format.	
`<opType>+`	Optional. One or more operation types for which you want to obtain the status.	
	DataService	Data service
	DatabaseOperation	Database operation
	HttpOperation	HTTP operation

avaki upgrade

Upgrade Avaki software on the specified grid servers and share servers.

Note See the *Sybase Avaki EII Administration Guide* for instructions on performing upgrades.

Syntax

```
avaki upgrade {--all | --server=<server-name>}
   [--version=<version>]
```

`--all | --server= <server-name>`	Upgrade the specified grid servers and share servers. Enter **avaki upgrade --all** to upgrade all grid servers and share servers in this Avaki domain, or specify the name of a single grid server or share server.
`--version= <version>`	Optional. Specify the version of Avaki software to be upgraded to. The default is the current version of the system code.

avaki upgrade --info

Display the upgrade version for the specified grid servers or share servers.

Syntax

```
avaki upgrade --info [--all | --server=<server-name>]
```

`--all | --server= <server-name>`	List upgrade information for the specified grid servers and share servers. Enter **avaki upgrade --info --all** to display upgrade information on all grid servers and share servers in this Avaki domain, or specify the name of a single grid server or share server.

avaki user

Enable or disable a grid user account on an LDAP or NIS authentication service.

To enable or disable a user account, you must be a member of the Administrators group or the UserAdministrators group.

Syntax

```
avaki user {<user-grid-path> | <qualified-user-name>}
  --enabled={true | false}>
```

`<user-grid-path> | <qualified-user-name>`

Specify the name of the user that you are enabling on the authentication service. You can specify the path to the data catalog directory of the user account. For example:

```
/System/Domains/Bedrock/Services/AuthServices/
Grid/DefaultAuthService/Users/wilma
```

Alternatively, you can specify the qualified user name using the following format:

```
<user>@<authservice>.<authservicetype>.<domain>
```

For example:

```
fred@slaterock.nis.bedrock
```

The elements of <qualified-user-name> are as follows:

`<user>`	The name of the user for which you want to obtain the user's name and contact information.
`<authservice>`	The name of the user's authentication service.
`<authservicetype>`	The type of authentication service (Grid, Nis, or Ldap).
`<domain>`	The name of the user's Avaki domain.

`--enabled={true | false}>`

Indicate whether the user account will be enabled (true) or disabled (false) on the authentication service.

avaki user --create

Create a grid user account.

To use this command, you must be a member of the Administrators group or the User-Administrators group, or you must have permission to create user accounts.

Syntax

```
avaki user --create [--real-name=<real-name>] [--email=<email>]
  <grid-user> <password>
```

`--real-name=<real-name>`	Optional. The full name of the user.
`--email=<email>`	Optional. The e-mail address of the user.
`<grid-user>`	The name of this user account.
`<password>`	The password for this user account. If you're executing this command on a Unix system and the password contains non-alphanumeric characters (! or &, for instance), enclose the password in single quotation marks to prevent the shell from interpreting the special characters.

avaki user --db-mapping --add

Add a database identity mapping for this user. A database identity mapping overrides the user's default user name and password.

To use this command, you must be a member of the Administrators group or the User-Administrators group.

Syntax

```
avaki user --db-mapping --add {<user-grid-path> |
  <qualified-user-name>} <qualified-dbconn-name> <db-username>
  <db-password>
```

`<user-grid-path> | <qualified-user-name>`

Specify the name of the user for which you are adding an identity mapping. You can specify the data catalog path to the directory of the user account. For example:

```
/System/Domains/Bedrock/Services/AuthServices/
Grid/DefaultAuthService/Users/wilma
```

Alternatively, you can specify the qualified user name using the following format:

```
<user>@<authservice>.<authservicetype>.<domain>
```

For example:

```
fred@slaterock.nis.bedrock
```

The elements of <qualified-user-name> are as follows:

`<user>`	The name of the user for which you want to obtain the user's name and contact information.
`<authservice>`	The name of the user's authentication service.
`<authservice-type>`	The type of authentication service (Grid, Nis, or Ldap).
`<domain>`	The name of the user's Avaki domain.

`<qualified-dbconn-name>`

A database connector on which per-user database identity mappings have been enabled. Before configuring a mapping for a user, enable database identity mappings on the database connector that connects to your database by creating an attribute for the database connector with the following parameters:

Name	`dbconn/UseIdentityMapping`
Type	`string`
Value	`true`

For information about creating new attributes, see **avaki attribute --update** (page 21).

The format for <qualified-dbconn-name> is:

```
[<domain-name>.]<database-connector-name>
```

For example:

```
Bedrock.sybase
```

The elements of <qualified-dbconn-name> are as follows:

`<domain-name>`	Optional. The grid domain where the database connector is deployed. The default is the current domain.
`<database-connector-name>`	The name of the database connector.

`<db-username>`

The database user name to map to `<user-grid-path>` or `<qualified-user-name>`.

`<db-password>`

The database password to map to `<user-grid-path>` or `<qualified-user-name>`.

avaki user --db-mapping --delete

Delete a database identity mapping for a user.

To use this command, you must be a member of the Administrators group or the User-Administrators group.

Syntax

```
avaki user --db-mapping --delete {<user-grid-path> |
  <qualified-user-name>} <qualified-dbconn-name>
```

`<user-grid-path> | <qualified-user-name>`

Specify the name of the user for which you are deleting the identity mapping. You can specify the data catalog path to the directory of the user account. For example:

```
/System/Domains/Bedrock/Services/AuthServices/
Grid/DefaultAuthService/Users/
wilma
```

Alternatively, you can specify the qualified user name using the following format:

```
<user>@<authservice>.<authservicetype>.<domain>
```

For example:

```
fred@slaterock.nis.bedrock
```

The elements of <qualified-user-name> are as follows:

`<user>`	The name of the user whose database identity mapping you want to delete.
`<authservice>`	The name of the user's authentication service.
`<authservice-type>`	The type of authentication service (Grid, Nis, or Ldap).
`<domain>`	The name of the user's Avaki domain.

`<qualified-dbconn-name>`

A database connector on which per-user identity mappings have been enabled. The format is:

```
[<domain-name>.]<database-connector-name>
```

For example:

```
Bedrock.sybase
```

The elements of <qualified-dbconn-name> are as follows:

`<domain-name>`	Optional. The grid domain where the database connector is deployed. The default is the current domain.
`<database-connector-name>`	The name of the database connector.

avaki user --db-mapping --list

List the database identity mapping for a user.

To use this command, you must be a member of the Administrators group or the User-Administrators group.

Syntax

```
avaki user --db-mapping --list {<user-grid-path> |
  <qualified-user-name>}
```

`<user-grid-path> | <qualified-user-name>`

Specify the name of the user for which you are listing the identity mappings. You can specify the data catalog path to the directory of the user account. For example:

```
/System/Domains/Bedrock/Services/AuthServices/
Grid/DefaultAuthService/Users/wilma
```

Alternatively, you can specify the qualified user name by using the following format:

```
<user>@<authservice>.<authservicetype>.<domain>
```

For example:

```
fred@slaterock.nis.bedrock
```

The elements of <qualified-user-name> are as follows:

`<user>`	The name of the user whose identity mapping you want to obtain.
`<authservice>`	The name of the user's authentication service.
`<authservice-type>`	The type of authentication service (Grid, Nis, or Ldap).
`<domain>`	The name of the user's Avaki domain.

avaki user --delete

Delete a user account from a grid domain.

To use this command, you must be a member of the Administrators or UserAdministrators group, or you must have permission to delete user accounts.

Syntax

```
avaki user {--delete | -d} {<users> | {--all | -a}}
```

`<users> | {--all | -a}`	The name of the user account to delete. To delete all users, specify --all or -a. To specify one or more users, enter a space-delimited list of users.

avaki user --info

Display a user's name and contact information.

Syntax

```
avaki user --info <qualified-user-name>
```

`<qualified-user-name>`	Specify the qualified name of the user account for which you want to obtain a full name and contact information. The format is:
	`<user>@<authservice>.<authservicetype>.<domain>`
	For example:
	`fred@slaterock.nis.bedrock`
	The elements of <qualified-user-name> are as follows:
	`<user>` — The name of the user whose contact information you want to obtain.
	`<authservice>` — The name of the user's authentication service.

<authservice-type>	The type of authentication service (Grid, Nis, or Ldap).
<domain>	The name of the user's Avaki domain.

avaki user --list-group

List all the groups to which a user belongs.

Syntax

```
avaki user {--list-group | -l} <qualified-user-name>
```

<user-grid-path> | <qualified-user-name>	Specify the qualified name of the user for which you want to obtain a group membership list. The format is:
	`<user>@<authservice>.<authservicetype>.<domain>`
	For example:
	`fred@slaterock.nis.bedrock`
	The elements of <qualified-user-name> are as follows:

<user>	The name of the user for which you want to obtain a group membership list.
<authservice>	The name of the user's authentication service.
<authservice-type>	The type of authentication service (Grid, Nis, or Ldap).
<domain>	The name of the user's Avaki domain.

avaki view --add-schedule

Add a generation schedule to a view.

To use this command, you must have write permission on the view.

Syntax

```
avaki view --add-schedule <view-path>
  {--periodic-schedule=<period-spec> |
  --cron-schedule=<cron-spec> |
  --one-time-schedule=<time-stamp>}
  [--start-time=<time-stamp>] [[--end-time=<time-stamp>] |
  [--max-iterations=<num-iterations>]]
  [--time-zone=<time-zone>] [--exclusion=<exclusion-name>]+
  [--exclusion-action={ drop | sooner | later }]
  [--reschedule-offset=<offset-spec>]
```

`<view-path>`
: The full grid path to the generated view (that is, the output file in which the view data is stored).

`--periodic-schedule=<period-spec>`
: Specify a periodic schedule for generating the view. Use the following syntax:

  ```
  --periodic-schedule="<number-of-units> <unit-type>"
  ```

`<number-of-units>`	An integer from 1 to 2147483648
`<unit-type>`	seconds, minutes, hours, days, weeks, or months

  In the following example, the periodic schedule is set to 30 seconds:

  ```
  --periodic-schedule="30 seconds"
  ```

`--cron-schedule=<cron-spec>`

Specify how often a cron-based scheduler for generating views generates the view. Use the following syntax:

```
--cron-schedule="<seconds> <minutes> <hours>
<days-of-month> <months> <days-of-week>
[<years>]"
```

For details about the <cron-spec> syntax, see Appendix C, "Configuring cron schedules".

`--one-time-schedule=<time-stamp>`

Specify a single time and date for generating the view. Use the following syntax:

```
--one-time-schedule="{<hh>:<mm>:<ss> {am | pm}
<mm>-<dd>-<yyyy> | now}"
```

`--start-time=<time-stamp>`

Specify the time and date that the view generator will start running, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> |
now}
```

`--end-time=<time-stamp>`

Specify the time and date that the view generator will stop running, in this format:

```
{<hh>:<mm>:<ss> {am | pm} <mm>-<dd>-<yyyy> |
now}
```

`--max-iterations=<num-iterations>`

Specify the number of times a schedule will repeat.

`<num-iterations>`	An integer from 1 to 2147483648. If no value is specified, the schedule repeats indefinitely.

`--time-zone=<time-zone>`

Specify the time zone for the schedule. The schedule can be specified according to the grid server's time zone or relative to Greenwich Mean Time (GMT). If no value is specified for the time zone, the server's local time zone is used. The format for the time zone is as follows:

GMT+12:00 Fiji	GMT+02:00 Cairo
GMT+11:00 Noumea	GMT+01:00 Paris
GMT+10:00 Sydney	GMT London

GMT+9:30 Adelaide	GMT-01:00 Azores
GMT+09:00 Tokyo	GMT-02:00 Mid-Atlantic
GMT+08:00 Hong Kong	GMT-03:00 Rio de Janeiro
GMT+07:00 Bangkok	GMT-03:30 Newfoundland
GMT+06:30 Rangoon	GMT-04:00 Caracas
GMT+06:00 Dacca	GMT-05:00 New York
GMT+05:45 Katmandu	GMT-06:00 Chicago
GMT+05:30 Calcutta	GMT-07:00 Denver
GMT+05:00 Karachi	GMT-08:00 San Francisco
GMT+04:30 Kabul	GMT-09:00 Juneau
GMT+04:00 Baku	GMT-10:00 Hawaii
GMT+03:30 Tehran	GMT-11:00 Samoa
GMT+03:00 Moscow	

`--exclusion=<exclusion-name>`

The name of an exclusion to this schedule. For information about creating schedule exclusions, see **avaki scheduleexclusion --create --daily** (page 201), **avaki scheduleexclusion --create --weekly** (page 205), **avaki scheduleexclusion --create --monthly** (page 203), **avaki scheduleexclusion --create --yearly** (page 207), and **avaki scheduleexclusion --create --custom** (page 199).

`--exclusion-action={drop | sooner | later}`

Specify the action to take if the view generator is scheduled to run during an exclusionary period.

drop	Don't run the data service during the exclusionary period.
sooner	Reschedule the data service to run before the exclusionary period.
later	Reschedule the data service to run after the exclusionary period.

`--reschedule-offset=<offset-spec>`	Specify the amount of time before or after the exclusionary period to run the view generator. Use the following syntax: `--reschedule-offset="<number-of-units> <unit-type>"` `<number-of-units>` An integer from 1 to 2147483648 `<unit-type>` Seconds, minutes, hours, days, weeks, or months

Example

This example adds a schedule in which the view pebblesmilestones.txt will be generated every 2 hours. In this example, the time zone is GMT-05:00 New York.

```
$ avaki view --add-schedule /GeneratedViews/
pebblesmilestonesview.txt --periodic-schedule="2 hours"
--time-zone="GMT-05:00 New York"
```

avaki view --create --database

Create a view generator that uses a database operation as its input source, then generate the view. To configure view properties such as a description that can be viewed by other users, use **avaki view --set-property** (page 275).

To use this command, you must be a member of the DataProviders group.

Syntax

```
avaki view --create --database <view-path>
  <qualified-dbop-name> [[<parameter-value>+]
  [--html | --csv | --stylesheet=<stylesheet-path>]
  [--engine={saxon | xalan | <class-name>}]]
  [--server=<grid-path>] [--no-generate]
  --run-user={<qualified-user-name>}
```

`<view-path>` — The full grid path to the generated view (that is, the output file in which the view data will be stored).

`<qualified-dbop-name>` — The database operation to use as the input source for the view. The format for the qualified name is:

```
[<domain-name>.]<database-connector-name>.
<database-operation-name>
```

For example:

```
Bedrock.sybase.listsalaries
```

The elements of <qualified-dbop-name> are as follows:

- `<domain-name>` — Optional. The grid domain where the database connector is deployed. The default is the current domain.
- `<database-connector-name>` — The name of the database connector that <database-operation-name> uses.
- `<dbop-name>` — The name of a database operation.

`<parameter-value>+`	Optional. One or more parameter values required by the target database operation. The parameter values will be processed in the order specified in the database operation's query. Any parameter value that includes a space must be enclosed in quotation marks. Any string at the end of the command that is enclosed in quotation marks will be interpreted as a single value. (That is, enclose each parameter value in its own pair of quotation marks.
`--html`	Optional. Produce a generated view in HTML format. (Use this option only when the input source is Avaki rowset XML, usually.)
`--csv`	Optional. Produce a generated view in comma-separated values format. (Use this option only when the input source is Avaki rowset XML, usually.)
`--stylesheet=<stylesheet-path>`	The grid path to a style sheet. Required if you specify a custom Java transform engine in the --engine option. If the input source is in XML, you can run an XSLT style sheet to convert the data to another XML schema, HTML, PDF, plain text, etc. In this case, use the --engine option to specify Saxon or Xalan, and use --stylesheet to provide the grid path to an XSLT style sheet. If you are using a custom Java transform engine, use --stylesheet to provide the grid path to a style sheet file that provides any needed information. (You must provide a style sheet file even if your transformation doesn't use it.)
`--engine={saxon | xalan | <class-name>}`	Optional. Specify the style sheet engine to use when generating this view. You can choose one of the two engines provided by Avaki, Saxon or Xalan, or you can write a custom transform engine in Java. If you write a custom engine, specify its TransformerFactoryImpl class name. You can find the name in your XSLT processor's JAR files. Here are two common ones: • For Saxon 7: net.sf.saxon.TransformerFactoryImpl • For jd.xslt: jd.xml.xslt.trax.TransformerFactoryImpl Default: saxon

`--server=<grid-path>`

Optional. Specify the name of the Avaki server on which the view will be created. For example, Bedrock. If you omit this option, the view is created on the current server.

`--no-generate`

Optional. Create a view generator but do not generate a view. (You can use **avaki view --generate** on page 274 to generate the view later.)

`--run-user={<qualified-user-name>}`

Specify the user to create the view as. Specify the qualified user name by using the following format:

```
<user>@<authservice>.<authservice-type>.<domain>
```

For example:

```
wilma@DefaultAuthService.Grid.Bedrock
```

The elements of <qualified-user-name> are as follows:

`<user>`	The name of the user to create the view as.
`<authservice>`	The name of the user's authentication service.
`<authservice-type>`	The type of authentication service (Grid, Nis, or Ldap).
`<domain>`	The name of the user's Avaki domain.

Example

```
$ avaki view --create --database /gridhome/fred/views/salaries
Bedrock.sybase.listsalaries
```

avaki view --create --data-service

Create a view generator that uses a data service as its input source, then generate the view. To configure view properties such as a description that can be viewed by other users, use **avaki view --set-property** (page 275).

To use this command, you must be a member of the DataProviders group.

Syntax

```
avaki view --create --data-service <view-path>
  [<qualified-ds-name> [<parameter-value>+] [--html | --csv |
  --stylesheet=<stylesheet-path>] [--engine={saxon | xalan |
  <class-name>}] [--server=<grid-path>] [--no-generate]
  --run-user={<qualified-user-name>}
```

`<view-path>`	The full grid path to the generated view (that is, the output file in which the view data will be stored).
`<qualified-ds-name>`	The data service to use as the input source for the view. The format for the qualified name is: `[<domain-name>.]<ds-name>`
`<parameter-value>+`	Optional. One or more parameter values required by the target data service operation. The parameter values will be processed in the order specified in the database service's query. Any parameter value that includes a space must be enclosed in quotation marks. Any string at the end of the command that is enclosed in quotation marks will be interpreted as a single value. (That is, enclose each parameter value in its own pair of quotation marks.)
`--html`	Optional. Produce a generated view in HTML format. (Use this option only when the input source is Avaki rowset XML, usually.)
`--csv`	Optional. Produce a generated view in comma-separated values format. (Use this option only when the input source is Avaki rowset XML, usually.)

`--stylesheet=<stylesheet-path>`	The grid path to a style sheet. Required if you specify a custom Java transform engine in the --engine option. If the input source is in XML, you can run an XSLT style sheet to convert the data to another XML schema, HTML, PDF, plain text, etc. In this case, use the --engine option to specify Saxon or Xalan, and use --stylesheet to provide the grid path to an XSLT style sheet. If you are using a custom Java transform engine, use --stylesheet to provide the grid path to a style sheet file that provides any needed information. (You must provide a style sheet file even if your transformation doesn't use it.)
`--engine={saxon | xalan | <class-name>}`	Optional. Specify the style sheet engine to use when generating this view. You can choose one of the two engines provided by Avaki, Saxon or Xalan, or you can write a custom transform engine in Java. If you write a custom engine, specify its TransformerFactoryImpl class name. You can find the name in your XSLT processor's JAR files. Here are two common ones: • For Saxon 7: net.sf.saxon.TransformerFactoryImpl • For jd.xslt: jd.xml.xslt.trax.TransformerFactoryImpl Default: saxon
`--server=<grid-path>`	Optional. Specify the name of the Avaki server on which the view will be created. For example, Bedrock. If you omit this option, the view is created on the current server.
`--no-generate`	Optional. Create a view generator but do not generate a view. (You can use **avaki view --generate** on page 274 to generate the view later.)

`--run-user={<qualified-user-name>}`

Specify the user to create the view as. Specify the qualified user name by using the following format:

```
<user>@<authservice>.<authservice-type>.<domain>
```

For example:

```
wilma@DefaultAuthService.Grid.Bedrock
```

The elements of <qualified-user-name> are as follows:

`<user>`	The name of the user to create the view as.
`<authservice>`	The name of the user's authentication service.
`<authservicetype>`	The type of the user's authentication service (Grid, Nis, or Ldap).
`<domain>`	The name of the user's Avaki domain.

avaki view --create --file

Create a view generator that uses a file or a generated view as its input source, then generate the view. To configure view properties such as a description that can be viewed by other users, use **avaki view --set-property** (page 275).

To use this command, you must be a member of the DataProviders group.

Syntax

```
avaki view --create --file <view-path> <file-path>
  [--stylesheet=<stylesheet-path>] [--engine={saxon | xalan |
  <class-name>}] [--server=<grid-path>] [--no-generate]
  --run-user={<qualified-user-name>}
```

`<view-path>`	The grid path to the output file in which the generated view will be stored.
`<file-path>`	The grid path to a file or generated view. This is the input source for the view generator.
`--stylesheet=<stylesheet-path>`	The grid path to a style sheet. Required if you specify a custom Java transform engine in the --engine option. If the input source is in XML, you can run an XSLT style sheet to convert the data to another XML schema, HTML, PDF, plain text, etc. In this case, use the --engine option to specify Saxon or Xalan, and use --stylesheet to provide the grid path to an XSLT style sheet. If you are using a custom Java transform engine, use --stylesheet to provide the grid path to a style sheet file that provides any needed information. (You must provide a style sheet file even if your transformation doesn't use it.)

`--engine={saxon | xalan | <class-name>}`

Optional. Specify the style sheet engine to use when generating this view. You can choose one of the two engines provided by Avaki, Saxon or Xalan, or you can write a custom transform engine in Java. If you write a custom engine, specify its TransformerFactoryImpl class name. You can find the name in your XSLT processor's JAR files. Here are two common ones:

- For Saxon 7: net.sf.saxon.TransformerFactoryImpl
- For jd.xslt: jd.xml.xslt.trax.TransformerFactoryImpl

Default: saxon

`--server=<grid-path>`

Optional. Specify the name of the server on which the view will be created. For example, Bedrock.

`--no-generate`

Optional. Create a view generator but do not generate a view. (You can use **avaki view --generate** (page 274) to generate the view later.)

`--run-user={<qualified-user-name>}`

Specify the user to create the view as. Specify the qualified user name by using the following format:

```
<user>@<authservice>.<authservice-type>.<domain>
```

For example:

```
wilma@DefaultAuthService.Grid.Bedrock
```

The elements of <qualified-user-name> are as follows:

`<user>`	The name of the user to create the view as.
`<authservice>`	The name of the user's authentication service.
`<authservicetype>`	The type of authentication service (Grid, Nis, or Ldap).
`<domain>`	The name of the user's Avaki domain.

Example

```
$ avaki view --create --file
/GeneratedViews/pebblesmilestonesview.txt
"/Shares/Water Buffalo Lodge/pebblesmilestones.txt"
--run-user=Wilma@DefaultAuthService.Grid.training
```

avaki view --delete

Delete a view.

To use this command, you must be a member of the Administrators group, or you must have delete permission on the view.

Syntax

```
avaki view --delete <view-path>
```

`<view-path>`	The grid path to the generated view file.

avaki view --delete-schedule

Delete a generation schedule from a view.

To use this command, you must have write permission on the view.

Syntax

```
avaki view --delete-schedule <view-path> <schedule-id>
```

`<view-path>`	The grid path to the generated view file.
`<schedule-id>`	The numeric ID of the schedule to delete. To determine a schedule's ID, use **avaki view --list-schedules** on page 275.

avaki view --depends

Show the dependencies to and from a view.

To use this command, you must have read permission on the message service on the grid server where the view is located, and you must have read permissions on the view.

Syntax

```
avaki view --depends <view-path>
```

`<view-path>`	The grid path to a generated view or view generator.

avaki view --garbage-collect

Force the garbage collector to run immediately on the specified view. The garbage collector frees up disk space by removing old results for the target view.

To use this command, you must be a member of the DataProviders group.

Syntax

```
avaki view --garbage-collect <view-path>
```

`<view-path>`	The grid path to the generated view or the view generator.

avaki view --generate

Regenerate a view.

To use this command, you must be a member of the DataProviders group.

Syntax

```
avaki view --generate <view-path>
```

`<view-path>`	The grid path to the generated view (if it already exists) or the view generator.

avaki view --info

Display a list of the properties of a generated view and their values. For descriptions of the properties that can be used to configure a view, see **avaki view --set-property** on page 275.

To use this command, you must be a member of the DataProviders group.

Syntax

```
avaki view --info <view-path>
```

`<view-path>`	The grid path to the generated view or the view generator.

avaki view --list-schedules

List the generation schedules for a view.

Syntax

```
avaki view --list-schedules <view-path>
```

`<view-path>`	The grid path to the generated view file.

avaki view --set-property

Configure the values for the properties that are used when a view is generated. Properties you can set include a description of the view, the style sheet for the view, parameters for the style sheet, and the schedule for the view. See the "Syntax" section, below, for details. To display current values of the properties, use **avaki view --info** (page 274).

To use this command, you must be a member of the DataProviders group.

Syntax

```
avaki view --set-property <view-path> <property-name>
  <property-value>*
```

`<view-path>`	The grid path to the generated view or the view generator.
`<property-name>`	The name of the property to configure. The property names are as follows:

General Properties

You can configure any of the following parameters for a file view generator, a data service view generator, or a database view generator:

description	Explain what this view is for, who owns it, when it is allowed to run, or other details that will be helpful to those who use it. You can display the description using **avaki view --info** (page 274).
gc-interval	The time in seconds between garbage collection cycles on expired generated views. Default: 1800 (half an hour) Range: 0 to 2147483647 seconds
gc-timeout	A timer (in seconds) that starts at the last access of an old generated view. When the timeout value is reached, the garbage collector deletes the old generated view. (A generated view is considered old when a newer version of the same view exists.) Set this property to 0 to cause old views to expire immediately upon the appearance of a newer version. Default: 3600 (1 hour) Range: 0 to 2147483647 seconds
gen-view-path	(Read-only) The grid directory in which the generated view will be stored.
notification-list	The grid path to one or more views that this view depends upon. When a specified view changes, the dependent view is regenerated. To specify more than one view, enter a space-delimited list of views.

output.xslt-engine	The class name of the TrAX transformer factory for the style sheet engine you want to use. Avaki includes two style sheet engines: • com.icl.saxon.TransformerFactoryImpl • org.apache.xalan.trax.TransformerFactoryImpl Default: com.icl.saxon.TransformerFactoryImpl
output.xslt-path	The full grid path to the style sheet that will be used to format the generated view.
output.xslt-params	One or more colon-separated <name>:<value> pairs, where <name> is a stylesheet property and <value> is its value. The name:value pairs are passed to the stylesheet. If you enter more than one name:value pair, separate the pairs with spaces.
schedule	The schedule or schedules under which this view will automatically run. Syntax: `"every <interval> <units> starting {<timestamp> | now}"+` <interval>: an integer from 0 to 2147483648 <units>: days, hours, or minutes <timestamp>: A date and time, in this format: yyyy-mm-dd hh:mm:ss {am | pm} To specify more than one schedule, enter a space-delimited list of schedules. To turn off the schedule, set the schedule to an empty string. See "Examples" on page 279.
server-path	(Read-only) The grid path to the Avaki server on which the generated view will be created.

Database View Generator Properties

You can configure any of the following parameters for a database view generator:

input.dbop	The name of the database operation to use as the input source for a database view generator.
input.input-file	The name of a file that contains data to load into the database. For information on the XML format required for input files, see the *Sybase Avaki EII Administration Guide*.
input.params	One or more parameters for a SQL query. To specify more than one parameter, enter a space-delimited list of parameters.
input.xml-format	Specify the XML schema for the view generator's input data. Default: default Valid values: default (Avaki's rowset format), oracle_xsu

File View Generator Properties

You can configure the following parameter for a file view generator:

input.file-path	The full grid path to the file to use as the input source for a file view generator.

`<property-value>*`	Specify a value for the property you are configuring. Property values are optional only for properties that take lists (multiple values).

Examples

This command makes the view /GeneratedViews/Pebbles dependent on the view /GeneratedViews/Dino so that /GeneratedViews/Pebbles is regenerated when /GeneratedViews/Dino changes.

```
$ avaki view --set-property /GeneratedViews/Pebbles
notification-list /GeneratedViews/Dino
```

This command changes the SQL parameters used for the database view SalesYTD to 2005.

```
$ avaki view --set-property /GeneratedViews/SalesYTD
input.params 2005
```

This command sets a view to run every 4 days starting at 1:59:59 PM on Sept. 3, 2005.

```
$ avaki view --set-property /GeneratedViews/Dino schedule
"every 4 days starting 2005-09-03 01:59:59 pm"
```

This command removes a view's previously configured schedule.

```
$ avaki view --set-property /GeneratedViews/Dino schedule
```

avaki view --update

Configure which user a view should be run as.

To use this command, you must be a member of the Administrators group, or you must have write permission on the view.

Syntax

```
avaki view --update <view-path>
  --run-user={<qualified-user-name> | null}
```

`<view-path>` — The grid path to the generated view file.

`--run-user={<qualified-user-name> | null}` — Specify the user to run the view as. Use this format:

`<user>@<authservice>.<authservicetype>.<domain>`

For example:

`wilma@DefaultAuthService.Grid.Bedrock`

The elements of <qualified-user-name> are as follows:

`<user>`	The name of the user to create the view as.
`<authservice>`	The name of the user's authentication service.
`<authservice-type>`	The type of authentication service (Grid, Nis, or Ldap).
`<domain>`	The name of the user's Avaki domain.

If you specify null, the view is run as the current user.

avaki virtualdatabase --allow-dbop-creation

Allow a user to create virtual database operations.

Syntax

```
avaki virtualdatabase --allow-dbop-creation
  <qualified-user-name>
```

`<qualified-user-name>`

Specify the user who can create virtual database operations. Use the following format:

`<user>@<authservice>.<authservicetype>.<domain>`

For example:

`wilma@DefaultAuthService.Grid.Bedrock`

The elements of <qualified-user-name> are as follows:

`<user>`	The name of the user who can create virtual database operations.
`<authservice>`	The name of the user's authentication service.
`<authservice-type>`	The type of authentication service (Grid, Nis, or Ldap).
`<domain>`	The name of the user's Avaki domain.

avaki virtualdatabase --disallow-dbop-creation

Prevent a user from creating virtual database operations.

Syntax

```
avaki virtualdatabase --disallow-dbop-creation
  <qualified-user-name>
```

`<qualified-user-name>`

Specify the user who must not create virtual database operations. Use the following format:

```
<user>@<authservice>.<authservicetype>.<domain>
```

For example:

```
wilma@DefaultAuthService.Grid.Bedrock
```

The elements of <qualified-user-name> are as follows:

`<user>`	The name of the user who is not allowed to create virtual database operations.
`<authservice>`	The name of the user's authentication service.
`<authservice-type>`	The type of authentication service (Grid, Nis, or Ldap).
`<domain-name>`	The user's grid domain.

avaki virtualdatabase --execute

Execute an ad-hoc query (a SQL statement) against the Avaki virtual database.

Syntax

```
avaki virtualdatabase --execute [--params={in | out |
  inout}:<parameter-types>] <sql> [<parameter-value>+]
```

`--params={in | out | inout}:<parameter-types>`

Optional. If the virtual database accepts input parameters or produces output parameters, use this option to specify the data type for each input and output. There is no default.

Use in for input parameters, out for output parameters, and inout for parameters that can be both. If you specify multiple parameters, use semicolons (;) to separate them.

You can use the following <parameter-types>:

BIGINT	LONGVARBINARY
BINARY	LONGVARCHAR
BIT	NUMERIC
BOOLEAN	ORACLE_CURSOR
BLOB	OTHER
CHAR	REAL
CLOB	SMALLINT
DATE	TIME
DECIMAL	TIMESTAMP
DOUBLE	TINYINT
FLOAT	VARBINARY
INTEGER	VARCHAR

`<sql>`

Enter the SQL statement that the virtual database will execute. Enclose the statement in quotation marks.

`<parameter-value>+`

Optional. One or more parameter values required by the target virtual database. If a database operation is associated with the virtual database, the parameter values will be processed in the order specified in the database operation's query. Any parameter value that includes a space must be enclosed in quotation marks. Any string at the end of the command that is enclosed in quotation marks will be interpreted as a single value. (That is, enclose each parameter value in its own pair of quotation marks.)

avaki virtualdatabase --show-tables

Display SQL views in the Avaki virtual database.

Syntax

```
avaki virtualdatabase --show-tables [--schema={DATASERVICE |
  VIRTUALDB | <database-connector-name>}]
  [--table=<table-name>] [--type={TABLE | VIEW}]
```

`--schema={DATASERVICE | VIRTUALDB | <database-connector-name>}`	Optional. Display SQL views derived from one of the following sources. If you omit this option, Avaki displays SQL views derived from all sources.	
	`DATASERVICE`	Display SQL views based on data services.
	`VIRTUALDB`	Display SQL views based on virtual database operations.
	`<database-connector-name>`	Display SQL views based on the specified database connector. Use the database connector name for both provisioned SQL views and SQL views generated from database operations.
`--table=<table-name>`	Optional. The name of a SQL view (table) to display.	
`--type={TABLE | VIEW}`	Optional. Display SQL views of one of the following types. If you omit this option, Avaki displays SQL views of both types.	
	`TABLE`	Display provisioned SQL views.
	`VIEW`	Display SQL views generated from database operations and data services.

Example

This command displays all the tables derived from virtual database operations. (There is only one such table in the current domain.)

```
$ avaki virtualdatabase --show-tables --schema=VIRTUALDB
```

```
====================== TABLE ======================
TABLE_NAME: VDOTESTINGWORKSTABLESTARTABLE
TABLE_CAT: mydomain.sybase.com
TABLE_SCHEM: VIRTUALDB
TABLE_TYPE: VIEW
REMARKS: null
TYPE_CAT: null
TYPE_SCHEM: null
TYPE_NAME: null
SELF_REFERENCING_COL_NAME: null
REF_GENERATION: null
=================== COLUMN ====================
COLUMN_NAME: title; DATA_TYPE: 12; TYPE_NAME: VARCHAR; COLUMN_SIZE: 128;
DECIMAL_DIGITS: 0; NUM_PREC_RADIX: 0; NULLABLE: 1; REMARKS: null; COLUMN_DEF:
null; CHAR_OCTET_LENGTH: 0; ORDINAL_POSITION: 1; IS_NULLABLE: YES; SCOPE_CATLOG:
null; SCOPE_SCHEMA: null; SCOPE_TABLE: null; SOURCE_DATA_TYPE: 0; PRECISION: 128;
=================== COLUMN ====================
COLUMN_NAME: guid; DATA_TYPE: 12; TYPE_NAME: VARCHAR; COLUMN_SIZE: 128;
DECIMAL_DIGITS: 0; NUM_PREC_RADIX: 0; NULLABLE: 1; REMARKS: null; COLUMN_DEF:
null; CHAR_OCTET_LENGTH: 0; ORDINAL_POSITION: 2; IS_NULLABLE: YES; SCOPE_CATLOG:
null; SCOPE_SCHEMA: null; SCOPE_TABLE: null; SOURCE_DATA_TYPE: 0; PRECISION: 128;
=================== COLUMN ====================
COLUMN_NAME: description; DATA_TYPE: 12; TYPE_NAME: VARCHAR; COLUMN_SIZE: 128;
DECIMAL_DIGITS: 0; NUM_PREC_RADIX: 0; NULLABLE: 1; REMARKS: null; COLUMN_DEF:
null; CHAR_OCTET_LENGTH: 0; ORDINAL_POSITION: 3; IS_NULLABLE: YES; SCOPE_CATLOG:
null; SCOPE_SCHEMA: null; SCOPE_TABLE: null; SOURCE_DATA_TYPE: 0; PRECISION: 128;
=================== COLUMN ====================
COLUMN_NAME: dp; DATA_TYPE: 12; TYPE_NAME: VARCHAR; COLUMN_SIZE: 128;
DECIMAL_DIGITS: 0; NUM_PREC_RADIX: 0; NULLABLE: 1; REMARKS: null; COLUMN_DEF:
null; CHAR_OCTET_LENGTH: 0; ORDINAL_POSITION: 4; IS_NULLABLE: YES; SCOPE_CATLOG:
null; SCOPE_SCHEMA: null; SCOPE_TABLE: null; SOURCE_DATA_TYPE: 0; PRECISION: 128;

Number of columns: 4

$
```

avaki virtualschema --deploy

Deploy a virtual schema (metadata) model into the data catalog.

Syntax

```
avaki virtualschema --deploy --schemaname=<virtual-schema-name>
  <local-path>
```

`--schemaname=<virtual-schema-name>`	The name of the virtual schema to deploy.
`<local-path>`	The local path to the virtual schema to deploy.

avaki virtualschema --undeploy

Undeploy a virtual schema (metadata) model from the data catalog.

Syntax

```
avaki virtualschema --undeploy
  --schemaname=<virtual-schema-name>
```

`--schemaname=<virtual-schema-name>`	The name of the virtual schema to undeploy.

avaki --whoami

Display your Avaki user name. Similar to the Unix **whoami** command.

Syntax

```
avaki whoami
```

– This command has no options.

Appendix A

Attribute reference

This appendix describes cache attributes that you can create as well as read-only system attributes that Avaki creates as needed. You can use the **avaki attribute** commands to create, modify, and delete attributes. See page 19 for details about **avaki attribute**.

This appendix covers the following types of attributes:

Avaki grid server attributes

This section describes read-only system attributes that provide information about an Avaki grid server.

Attribute	Description
system/ObjectHostName	The DNS name or IP address of the machine on which the object is running.
system/os.name	The name of the operating system on the machine on which the object is running.
system/os.arch	The architecture of the machine on which the object is running.

File attributes

This section describes read-only system attributes that provide information about files in a data grid.

Attribute	Description
system/FileSize	The size of a file (in bytes).
system/ChangeTime	The time a file was last changed.
system/ModificationTime	The time a file was last modified.
system/LastAccessTime	The time a file was last accessed.
system/BlockSize	The size of a file's blocks.
system/Blocks	The number of blocks to a file.

Cache service attributes

By default, the attributes for a cache service are configured on a cache-wide basis (for more information, see the *Sybase Avaki EII Provisioning and Advanced Data Integration Guide*). If you set the following attributes on a per-file basis, the settings override the cache-wide read-only system attributes.

Attribute	Description
cacheable/CoherenceWindow	Integer specifying the time in seconds after the content has last been updated during which the proxy cache service assumes the content to be fresh and does not check back with the source for expiration. Set this attribute to –1 for a static file that should never expire. This attribute overrides the value of the proxy cache service's default coherence window setting, which configures the coherence window on a cache-wide basis. Valid values: –1, 0 to 2147483647 seconds
cacheable/OfflineExpiration	Integer specifying the time in seconds for which content is allowed to remain valid after its source is determined to be offline. A value of –1 means that this file is always available for offline access when its source is determined to be unreachable. This attribute overrides the value of the proxy cache service's offline expiration setting, which configures the coherence window on a cache-wide basis. Valid values: –1, 0 to 2147483647 seconds

Appendix B

Setting up DGAS properties files

This appendix describes the properties file used by Avaki data grid access servers (DGASes).

About the properties file

If you are starting and configuring a data grid access server using the command line tools, you have the option to use a properties file. The DGAS's properties file lets you specify startup options for the DGAS, including NFS and CIFS behavior, the location and behavior of the caches, and threading options. You can also specify these options in the web UI, but you cannot use a properties file if you set up a DGAS in the web UI.

Store the properties file in a file system where it is accessible to the DGAS. You can store it under any path and name; the default path is the Avaki installation directory. There is no default file name. Our convention is to use the name dgas.properties.

The section that follows includes a sample properties file. To display a list that includes the names, descriptions, and default values of all DGAS properties, use a command of this form:

```
avaki dgas startup <dgas-name>
```

Properties file format

Below is a sample of the properties file that sets startup options for a data grid access server. The list of properties changes frequently; for authoritative descriptions, use the following command, in which <dgas-name> is the name of a data grid access server:

```
avaki dgas --get-property-list <dgas-name>
```

Read this sample file as a format guideline. Because this sample would set most properties to their default values, it would have no effect if you used it with your DGAS.

Notes Several property values shown in this sample file are not literal defaults:

- The db-path and cache-path properties require paths as values. The paths in this sample are for a default Windows installation. For a Unix installation, replace the string "C:\Avaki Data Grid 6.0" with the path to the Avaki installation directory on the DGAS machine, and use slashes instead of backslashes in the path.
- The default value of the cifs-server-name property is the name of the machine on which the DGAS is running.

Lines in the sample file that begin with # are comments.

```
##################################################################
# Optional properties file for Avaki Data Grid Access Servers
#
##################################################################
non-idempotent-ops-cache-ttl=10000
lock-manager-enabled=false
cifs-wins-server=WINS_SERVER_NAME
cache-block-size=104857600
inactive-blocks-high-water=-1
precreate-async-write-threads=false
cifs-enabled=false
unix-chown-semantics=true
cifs-max-dirent-per-read=100
cache-frags-per-block=100
inactive-blocks-low-water=31457280
user-cache-size=5
unix-file-mode-semantics=false
cache-enabled=true
cifs-bind-address=*
nfs-preferred-xfer-size=32768
nfs-v3-enabled=true
```

```
non-idempotent-ops-cache-size=20
name=Access1
nfs-max-dirent-per-read=50
async-write-blocks-pool-size=20
max-async-write-bsize=1048576
updates-portmap=true
db-path=C:\Avaki Data Grid 6.0\DGAS\dgas_db
cifs-name-registration-mode=1
max-num-async-write-blocks=5
max-async-write-threads=6
mem-cache-stages=2
cache-path=C:\Avaki Data Grid 6.0\DGAS\dgas_caches
min-async-write-threads=0
file-cache-coherence-window=60000
dir-cache-size=100
nlm-port=0
max-cached-files=50
nfs-port=2049
mount-port=0
min-nfs-udp-threads=0
cifs-server-name=<DGAS host name>
cifs-host-announce-interval=5
netbios-smb-enabled=false
user-mapping-ttl=7200000
tcpip-smb-enabled=false
num-active-blocks=10
mount-server-enabled=true
case-sensitive-searches=false
precreate-nfs-udp-threads=false
mem-fetch-threshold=1048576
dir-cache-coherence-window=120000
file-obj-cache-size=256
cifs-name-registration-frequency=3600
nfs-v2-enabled=true
cifs-server-comment=Avaki CIFS Server
nfs-max-dirplusent-per-read=50
cifs-server-broadcast-address=255.255.255.255
dir-cookie-check-enabled=false
cifs-server-domain=AVAKI
max-nfs-udp-threads=10
```

Appendix C

Configuring cron schedules

You can use a cron specification to indicate the schedule by which the following tasks will be performed:

- Adding a caching schedule for a data service; see **avaki dataservice --add-schedule** (page 48).
- Adding a caching schedule for a database operation; see **avaki dbop --add-schedule** (page 74).
- Adding a pin schedule for a cached directory; **avaki directory --add-schedule** (page 118).
- Tagging a directory or a directory tree for caching within a cache service; see **avaki directory --cache** (page 122).
- Adding a pin schedule for a cached file; see **avaki file --add-schedule** (page 131).
- Tagging a file within a cache service for active caching; see **avaki file --pin** (page 138).
- Adding a schedule by which LDAP users will be automatically imported; see **avaki ldap --add-schedule** (page 150).
- Adding a schedule by which NIS users will be automatically imported; see **avaki nis --add-schedule** (page 178).
- Adding a rehash schedule to a shared directory; see **avaki share --add-rehash-schedule** (page 232).
- Adding a generation schedule to a view; see **avaki view --add-schedule** (page 260).

You can include a *cron schedule option* in any of the commands listed above. The syntax for a cron schedule option is as follows:

```
--cron-schedule="<seconds> <minutes> <hours>
<days-of-month> <months> <days-of-week> [<years>]"
```

Alternatively, you can include a *cron expression* in any advanced schedule in the web user interface. A cron expression's syntax is similar to that of the cron schedule option:

```
<seconds> <minutes> <hours> <days-of-month> <months>
<days-of-week> [<years>]
```

You can use the following values in either type of cron specification:

Cron field	Valid values	Special characters allowed*
<seconds>	An integer from 0 to 59	Comma (,), hyphen (-), asterisk (*), and slash (/)
<minutes>	An integer from 0 to 59	Comma (,), hyphen (-), asterisk (*), and slash (/)
<hours>	An integer from 0 to 23	Comma (,), hyphen (-), asterisk (*), and slash (/)
<days-of-month>	An integer from 1 to 31	Comma (,), hyphen (-), asterisk (*), slash (/), question mark (?), and L
<months>	An integer from 1 to 12 or JAN-DEC	Comma (,), hyphen (-), asterisk (*), and slash (/)
<days-of-week>	An integer from 1 to 7 or SUN-SAT	Comma (,), hyphen (-), asterisk (*), slash (/) question mark (?), L, and pound sign (#)
<years>	Optional. An integer from 1970 to 2099	Comma (,), hyphen (-), asterisk (*), and slash (/)

* See the table on the next page for more information on special characters.

You can use the following special characters in either type of cron specification:

Character	Description
Comma (,)	A comma may be used to specify additional values. For example, “MON,WED,FRI” in the day-of-week field means “the days Monday, Wednesday, and Friday.”
Hyphen (-)	A hyphen may be used to specify ranges. For example, “10-12” in the hour field means “the hours 10, 11, and 12.”
Slash (/)	A slash may be used to specify increments. For example, “0/15” in the seconds field means “the seconds 0, 15, 30, and 45.” The syntax “5/15” in the seconds field means “the seconds 5, 20, 35, and 50.”
Asterisk (*)	An asterisk can be used to specify all valid values in a given field. For example, “*” in the minute field means “every minute.”
Question mark (?)	A question mark can be used in the day-of-month and day-of-week fields to specify “no specific value.” This character is useful when you need to specify something in one of the two fields, but not in the other.
L	The L character may be used in the day-of-month and day-of-week fields. This character stands for “last” and has a different meaning in each of the two fields. For example, the value “L” in the day-of-month field means “the last day of the month”; day 31 for January, and day 28 for February on non-leap years. If used in the day-of-week field by itself, L means “7” or “SA.”. If used in the day-of-week field after another value, L means “the last xxx day of the month.: For example, “6L” means “the last Friday of the month.”
Pound sign (#)	A pound sign may be used in the day-of-week field to specify “the nth” xxx day of the month. For example, the value of “6#3” in the day-of-week field means the third Friday of the month (day 6 = Friday and “#3” = the 3rd one in the month).

Appendix D

Data type mapping files

When you provision a table into Avaki from a relational DBMS (that is, when you provision a *SQL view*), Avaki uses a default set of mappings between the SQL data types used in the DBMS and the data types used by Avaki's Axion query engine. If the default data type mappings yield unsatisfactory results, you can override them for a particular database connection. To override the default mappings, you must

- Create an XML mapping file as described in this appendix and share it into the Avaki data catalog.
- Use the **--mapping-file** option of the **avaki dbconn --jdbc** command (page 66) to specify the location of the mapping file.

In this appendix:

Format and contents of type mapping files

In your mapping file, specify only the subset of types that need to be mapped differently from the default.

The format of the mapping file is as follows:

```
<mappings>

   <mapping>
      <tableType>all</tableType>
      <sqlTypeInt>3</sqlTypeInt>
      <sqlTypeName>decimal</sqlTypeName>
      <outputType>long</outputType>
      <precisionSource>precision</precisionSource>
      <scaleSource>decimalDigits</scaleSource>
      <displaySizeSource>size</displaySizeSource>
   </mapping>

   <mapping>...</mappings>...

<mappings>
```

Each mapping in the file contains the following information. Elements are required except as noted.

Element	Valid values	Description

`tableType`	`all`
	`provisioned`
	`generated`
		To specify different mappings for SQL views (tables) that are provisioned into Avaki and SQL views that are generated from database operations or data services, enter a `mapping` element for each table type. Otherwise, enter a single mapping element for the designated type and set `tableType` to `all`.

Element	Valid values				Description
sqlTypeInt	ARRAY	2003	JAVA_OBJECT	2000	The integer constant representing the SQL type to be mapped. For example, use 3 for a decimal. See "Setting the source data type: data type inconsistencies" on page 304 for more information on setting the value of this element.
	BIGINT	-5	LONGVARBINARY	-4	
	BINARY	-2	LONGVARCHAR	-1	
	BIT	-7	NULL	0	
	BLOB	2004	NUMERIC	2	
	BOOLEAN	16	OTHER	1111	
	CHAR	1	REAL	7	
	CLOB	2005	REF	006	
	DATALINK	70	SMALLINT	5	
	DATE	91	STRUCT	2002	
	DECIMAL	3	TIME	2	
	DISTINCT	2001	TIMESTAMP	93	
	DOUBLE	8	TINYINT	-6	
	FLOAT	6	VARBINARY	-3	
	INTEGER	4	VARCHAR	12	
sqlTypeName	String				The name of the SQL type to be mapped. Specify the string the underlying database or database operation returns as the name of the SQL type.
outputType	boolean		varchar2		The Axion type to which you are mapping the designated SQL type. More information about Axion data types is available at **http://axion.tigris.org/datatypes.html**
	byte		float		
	unsignedbyte		number		
	short		numeric		
	unsignedshort		clob		
	integer		compressedclob		
	int		blob		
	smallint		compressedblob		
	unsignedinteger		date		
	long		time		
	character		timestamp		
	char		java_object		
	varchar		varbinary		

Element	Valid values		Description
precisionSource	charOctetLen dataType decimalDigits default isNullable name nullable numPrecRadix ordinalPos	precision remarks scopeCatalog scopeSchema scopeTable size sourceDataType typeName	Optional. The element in the native table's metadata that you are mapping to the precision element in Avaki.
scaleSource			Optional. The element in the native table's metadata that you are mapping to the scale element in Avaki.
displaySizeSource			Optional. The element in the native table's metadata that you are mapping to the displaySize element in Avaki. **Note:** Because Avaki does not allow displaySize to be smaller than the default for its data type, mapping the Avaki displaySize element to a SQL element with a smaller value has no effect.

Setting the source data type: data type inconsistencies

When a back-end database responds to a JDBC request, it sometimes performs type translation. As a result, a column that appears to be of one type when viewed or accessed through a JDBC-based tool or application (the SQuirreL SQL client, for example) might appear to be of a different type when viewed with a non-JDBC tool. This is not a common occurrence. However, when you set the value of the sql-TypeInt element in a mapping file, be aware that the database's internal notion of a column's data type is not important for mapping purposes—the type you need to capture is the one the database returns through JDBC.

Logging of type mapping

By default, Avaki's log4j logging mechanism records type mapping decisions during the creation of provisioned or generated SQL views (virtual tables). The log entries indicate the JDBC type being mapped to the Avaki/Axion type.

Logging of type mapping is controlled by a log4j category called `TypeMapping`. If you want the type mapping for a particular SQL view to be logged (or not logged), logging of type mapping must be enabled (or disabled) in the log properties file on the grid server that hosts the database operation or data service on which the SQL view is based.

For information on configuration of logging, see the appendix on log properties files in the *Sybase Avaki EII Administration Guide*.

Glossary

Terms printed in *italics* are defined in the glossary.

access control list
(ACL) A list, for a given file, directory, or other Avaki object, of permissions—read, write, execute, delete, and owner—that control which users and groups can view, modify, invoke, and remove the object, and edit the object's ACL.

ACL
See *access control list*.

ad-hoc query
A mechanism that lets you directly query a database in SQL. The query must run through an existing Avaki *database connector*. You can run an ad-hoc query using either the CLI or a *JDBC driver*. Ad-hoc queries can be thought of as single-use *database operations*.

attribute
A property of an *Avaki directory*, file, *service*, or other object. Each attribute has a name, a type (string, integer, float, date, time, or timestamp) and a value. System attributes are read-only; you can change the values of other attributes. You can also create new attributes and add them to objects as needed.

authentication service
A *service* associated with an *Avaki domain* that authenticates an Avaki user's identity and provides security credentials each time the user logs in. Avaki can be configured to use third-party directory services as authentication services for login; for user accounts created directly in the Avaki domain, Avaki uses its own default authentication service.

Avaki directory

Avaki software creates a single, unified namespace that is accessible (subject to Avaki *access control lists*) to all users in the *Avaki domain*. The namespace, called the *data catalog*, is arranged as a hierarchy of Avaki directories (folders). The catalog directory structure is stored by the domain's grid servers and its GDC, while the physical files remain in their original locations in your local file systems. When you work with directories, it's important to distinguish between Avaki directories, which are part of the data catalog, and local directories, which reside in your local file system.

Avaki domain

The basic administrative unit of the Avaki EII system. An Avaki domain consists, at a minimum, of one *grid domain controller* and may also include one or more *grid servers*, *share servers*, *proxy servers*, *data grid access servers*, and *command clients*. See also *domain name*.

Avaki group

A set of users who have the same permissions on one or more Avaki objects. You can use the group name in place of a user name when you set permissions or create *access control lists*.

Avaki installation directory

The directory in your local file system where Avaki software is installed. This is not a *data catalog* directory.

Avaki share

(Also shared directory.) A pointer in the Avaki *data catalog* to a directory or file in the underlying local file system. When you browse the data catalog, Avaki shares look like—and can be accessed like—other Avaki directories. Contrast with *CIFS share.*

Avaki server

A *service* that starts, stops, and monitors other Avaki services on a particular computer. Every server is part of an *Avaki domain*. A server is permanently attached to the computer where it is started. There are several types of server: *data grid access servers*, *grid domain controllers*, *grid servers*, *share servers*, and *proxy servers*.

Avaki Studio

A graphical, metadata-based data integration tool that lets you

- Build data flows by dragging and dropping input sources, operators, and output targets. You can deploy your data flows as Avaki *data services*.
- Import or create *metadata models* and apply them to Avaki objects or use them to build new data services.

In addition, you can use Studio to perform provisioning tasks (creating *database connectors*, *database operations*, *virtual database operations*, and *SQL views*), manipulate *categories*, and edit *ad-hoc queries* and *attributes*.

cache service

(Formerly proxy cache service.) A staging service that stores copies of files, *database operation* results, and *data service* results. Caching improves retrieval performance. To ensure that an object is stored in the cache, you can *pin* a file or directory in the data catalog, or schedule a database operation or data service. A cache service can provide remote caching, local caching, or both. The freshness of cached data is controlled by a data expiration interval that determines how long cached data is considered valid and by a cache coherence window that tells the cache service how often to check whether cached data is still valid. If cached data is too old to satisfy a new request (or is not stored in this cache), the cache service does one of the following:

- If the database operation or data service that produced the data is local to this cache service, the cache service triggers execution of the database operation or data service.
- If the database operation or data service that produced the data is remote from this cache service, this cache service requests the data from the data source's local cache service.

A cache service can be associated with a *data grid access server*, a *grid server*, or a local user in a CLI session. See also *local cache*, *remote cache*, *on-demand caching*, and *scheduled caching*.

category

A mechanism for classifying and organizing the contents of the *data catalog*. Like *Avaki directories*, categories serve as containers for objects in the data catalog. Anything in the data catalog—views, data services, shared files, even Avaki directories themselves—can be assigned to a category. Categories are hierarchical, they have attributes, and Avaki *access control lists* regulate access to them.

CIFS client

A machine that mounts files or directories from the Avaki *data catalog* by connecting to a *CIFS share* through an Avaki *data grid access server*. A CIFS client need not have Avaki software installed. (CIFS—Common Internet File System—is a file-sharing protocol based on the file system implemented by Windows.)

CIFS share

A directory or file that has been exported (shared) from the Avaki *data catalog*. A CIFS share can be mapped into a Windows file system like a network drive. When you browse the Windows file system, CIFS shares look like—and can be accessed like—other files and directories. CIFS shares are created through a *data grid access server*. Contrast with *Avaki share*.

client
Avaki supports several types of client: *Avaki Studio*, *CIFS clients*, *command clients*, JDBC/ODBC clients, *NFS clients*, *web clients, and WS clients*.

command client
A machine that can issue Avaki commands but does not contribute resources to the *Avaki domain*.

connect port
The connect port on a *grid domain controller*, *grid server*, *data grid access server*, *proxy server*, or *share server* accesses the JNDI naming service or RMI registry for the underlying application server. The connect port is one of many ports that a GDC or server uses to communicate with other Avaki objects. You must supply the connect port number of a target grid server or GDC whenever you connect a new object (another server, a copy of Avaki Studio, or a *command client*, for example) to an *Avaki domain*. When you *interconnect* two Avaki domains, you must supply each domain's connect port number to the other one.

data catalog
A hierarchical structure similar to a file system that encompasses all objects in an *Avaki domain*. The data catalog contains *Avaki directories* and files, *Avaki shares*, *Avaki servers*, *SQL views*, *database operations* and *data services*, and other objects.

data grid access server
(DGAS) An *Avaki server* that makes *Avaki directories* and their contents available to *CIFS clients* and *NFS clients*.

data service
An operation that transforms data obtained from sources in the *data catalog*. Input data can come from any number of sources, including:

- other data services
- data catalog files (which can be *generated views*)
- Avaki *database operations* (which in turn extract the data from relational databases)
- HTTP requests
- Web service invocations

You can generate the code that manipulates the data by creating a *view model* in *Avaki Studio*, or by writing a custom *data service plug-in* using Java, JavaScript, or XSLT. Data service output can be in rowset or XML format. Data services are run by the *execution services* on *grid servers*, they can be scheduled, and their results can be cached.

data service plug-in
The logic for a *data service*, written in Java, JavaScript, or XSLT. Data service plug-ins are modular—you can use the same plug-in for multiple data services. *Avaki Studio* creates data services and plug-ins simultaneously, so if you use Avaki Studio to create data services, you don't have to worry about plug-ins. You can also use the Avaki Plug-in Wizard to create data service plug-ins.

database connector
A mechanism that enables one or more *database operations, SQL views*, or *ad-hoc queries* to connect to a relational database.

database operation
(DBOP) A mechanism that can

- extract data from a relational database and deliver it on demand to a *view generator* or a *data service*, or
- modify data in a relational database.

A database operation can be a SQL statement or a stored procedure call.

dependency
A relationship in which an Avaki object requires input from other Avaki objects. A *data service* might require input from one or more *database operations* or from other data services. A *view generator* might depend on a database operation for input. A database operation can serve as an input source for one or more data services or view generators. Generated *SQL views* depend on database operations, virtual database operations, or data services. You can use *Avaki Studio*, the web UI, or the CLI to list input and output dependencies for any data service, database operation, or view.

DGAS
See *data grid access server*.

distributed transaction
A set of related operations (typically SQL operations such as SELECT, INSERT, UPDATE, DELETE, and CALL) that

- involve one or more databases, and
- might lead to unwanted results (such as leaving participating databases in an inconsistent state or producing inconsistent reads) if some of the operations complete and others do not, and therefore
- must all be executed at once, as a single transaction.

The individual operations that make up a distributed transaction are performed by *database operations* that use *database connectors* configured with XA-capable *JDBC drivers*; all the database operations are executed, using the two-phase commit protocol, by a specially configured *data service*. The two-phase commit protocol is designed to ensure that the participating databases will be left in a consistent state—that is, that all the operations in the distributed transaction will be completed, or none of them will.

domain name

A unique alphanumeric identifier for an *Avaki domain*. The domain name is assigned by the Avaki administrator when the Avaki domain is initialized. The domain name has a maximum length of 30 characters.

enterprise information integration

(EII) A software system that

- enables applications and users to access, without replication, both raw and integrated data from multiple heterogeneous distributed data sources while hiding the complexity of the data sources, and
- provides tools enabling users and data owners to further integrate and transform data.

exclusion

See *schedule exclusion*.

execution service

Execution services execute *data services*. There is an execution service on every *grid server*, and you can configure a pool of execution services for load-sharing. When a pool is in place, a data service can be run by any execution service in its grid server's pool.

failover

The transition of control from a failing or unreachable primary *grid domain controller* to a secondary grid domain controller.

federated data access

A scheme that allows independently controlled elements to be shared into a single namespace. Files, user accounts, and other objects maintain their separate identities and remain under the control of their owners, but—subject to access controls—the objects can be accessed, managed, and viewed as if they were part of a single system.

GDC

See *grid domain controller*.

generated view
A file created by a *view generator;* it may contain data obtained from a *database operation*, a *data service*, a file, or an HTTP source. Like other files, generated views exist in a local file system and are shared into the *data catalog*.

grid
A heterogeneous group of networked resources that appears and functions as one operating environment. A data grid like the Avaki Enterprise Information Integration (EII) system provides secure, shared access to data.

grid directory
See *Avaki directory*.

grid domain
See *Avaki domain*.

grid domain controller
(GDC) The first server in an *Avaki domain* is the grid domain controller. The GDC maintains a portion of the Avaki domain's namespace and provides authentication services. It can also run Avaki commands, share data, and monitor other servers. (That is, the GDC functions as a *grid server*.) If the domain is configured for *failover*, it has both a primary GDC and a secondary GDC; the secondary is updated at regular intervals and takes over management of the domain if the primary fails. Any Avaki shares managed by the primary are read-only on the secondary.

grid server
An *Avaki server* that maintains a portion of the *Avaki domain*'s namespace, runs Avaki services such as shares, execution services, caches, and searches, and allows you to run Avaki's web UI and execute Avaki commands.

group
See *Avaki group*.

hard link
Provides an alternate name for an item in the *data catalog*. Changes to the object's other names have no effect on the hard link: you can move or change a file's original name and the hard link will still know where to find the file. To delete a hard-linked object, you must remove its original name. Contrast with *soft link*.

interconnect
To create a unidirectional link from one *Avaki domain* to another. Interconnecting lets an Avaki domain make its *data catalog* visible to users in another domain (subject to Avaki access controls).

JDBC driver
JDBC (Java Database Connectivity) drivers allows application programmers to access database data shared in the *data catalog*. When a JDBC driver accesses data, it returns a JDBC result set that's immediately available to your program. JDBC drivers can:

- Call any *data service* in the data catalog
- Call any *database operation* in the data catalog
- Perform SQL `select` operations against *SQL views* in the data catalog

Sybase offers three JDBC drivers for use with Avaki EII software:

- The included Avaki JDBC driver
- jConnect, Sybase's standard JDBC driver
- An XA-capable driver for use with *database connectors* that support *distributed transactions*

link
See *hard link*, *soft link*.

local cache
A *cache service* that runs on the same *grid server* as a *database operation* or a *data service* that generates cachable data. The local cache stores results produced by local database operations and data services so they don't have to execute for every new request. See also *remote cache*.

metadata model
A construct in *Avaki Studio* that expresses a schema by defining a set of tables. A table in a metadata model can be mapped (linked) to an Avaki object such as a *data service* or a *database operation*, or to a table in a relational database. The mapping lets you address each mapped object by the name of the corresponding table in the metadata model. You can also derive a *view model* schema from a metadata model. When you do this, you ensure that the results of any data service deployed from the view model will conform to the metadata model's schema.

NFS client
A machine that mounts the Avaki *data catalog* (or a portion of it) as a directory by connecting to an Avaki *data grid access server*. An NFS client need not have Avaki software installed. (NFS—Network File System—lets you add file systems located on a remote computer to the directory structure on your own computer.)

ODBC
ODBC (Open DataBase Connectivity) is an API for databases on Windows. An ODBC driver (such as the the Sybase Organic ODBC driver included with Sybase ASE) allows Avaki to communicate with Windows database applications.

on-demand caching
A scheme by which an object is cached only if it's used—for example, results are cached when a *database operation* or a *data service* is executed, or a file is cached when a user or application reads it. On-demand caching uses a fixed expiration interval to determine data freshness. On-demand caching is suitable for objects that are rarely accessed or that change at irregular intervals. Contrast with *scheduled caching*.

pin
To mark an *Avaki directory* or file for *scheduled caching*. See also *cache service*.

plug-in
See *data service plug-in*.

primary GDC
See *grid domain controller*.

proxy server
An *Avaki server* that allows *Avaki domains* on opposite sides of a firewall or a Network Address Translator (NAT) to communicate with one another.

queries
See *ad-hoc query*.

query engine
An Avaki *service* that executes SQL queries against the *SQL views* (tables) that make up the Avaki *virtual database*. A query engine analyzes queries, pushes as much of the work as possible down to the underlying relational database (if there is one), and performs the remaining operations (such as joins across tables from different databases) itself. There is a query engine on each *grid server*.

remote cache
A *cache service* that runs on a grid server that is remote from an Avaki service (a *database operation* or a *data service*) that generates cachable data. The remote cache stores results produced by distant services so the results don't have to be fetched over the network to satisfy every new request. Users and applications that access remote data through the cache may have access to cached copies even when the remote data source is unavailable. See also *local cache*.

scheduled caching
A scheme by which an object is cached according to a schedule that you create. The schedule specifies when the object is first cached and how often (or following what trigger event, such as a change to a file) the cache is refreshed. If the object is a *data service* or a *database operation*, the schedule runs it to put fresh results in the cache. Scheduled caching, which overrides other types of caching, is suitable for objects that are updated frequently or on a regular basis. Contrast with *on-demand caching*.

schedule exclusion
A named period of time during which scheduled activities can be prevented from running. You can apply an exclusion to as many schedules as you want. Scheduled activities include refreshing *Avaki shares* and imported user accounts, and caching files, directories, and the results of *database operations*, *data services*, and *generated views*.

secondary GDC
See *grid domain controller*.

service
An Avaki object that performs a function in the domain (stores data or authenticates users, for example). Services provided in Avaki software include *Avaki directories*, *Avaki shares*, *Avaki servers*, *authentication services*, *execution services*, and user accounts.

share
A point of connection between the Avaki *data catalog* and a native file system or file system tool. Avaki supports two kinds of shares: *Avaki shares* and *CIFS shares*.

share server
An *Avaki server* whose only task is to manage *Avaki shares*—local directories that are exported (shared) into the *data catalog*. (Grid servers can also manage shares.)

shared directory
See *Avaki share*.

soft link
A pointer to a particular location (name) in the Avaki *data catalog*. If the object at that location is moved, deleted, or renamed, the soft link leads nowhere. Soft links can be created only in the CLI. Contrast with *hard link*.

SQL view
A virtual table—a *data catalog* entry that represents a table in a relational database, a *database operation*, or a *data service*. SQL views can be created in three ways:

- Provisioned directly from a table in an underlying database
- Generated from a database operation or data service
- Mapped from a database table, a database operation, or a data service, using the *Avaki Studio* metadata model editor

Every SQL view is part of the Avaki *virtual database*. SQL views are treated as relational tables by the Avaki *query engine*. SQL view data can be accessed using standard SQL statements by connecting to Avaki with ODBC or JDBC, or via an Avaki *virtual database operation*.

update notification

A message issued when a *generated view* is updated. A view that receives data from another view can be configured to regenerate itself (using the new data) upon receipt of an update notification.

view generator

A mechanism that does one of the following: extracts data from a file or an HTTP source, obtains data from an Avaki *data service*, or uses an Avaki *database operation* to extract data from a relational database. The view generator can display the data, perform an XSLT transform, save the data as a *generated view* file, and/or update a database. Contrast with *data service*.

view model

The graphical representation of a data flow that you can build in *Avaki Studio*. A view model typically includes one or more input sources (such as *database operations* or *data services*), one or more operations to combine or transform the data, and an output target. When you deploy a view model, it becomes an Avaki data service.

virtual database

The set of all *SQL views* in an *Avaki domain*, including those provisioned from external databases and those generated from *data services* and *database operations*. You can execute SQL queries on the SQL views in the virtual database as if they were tables in a single database.

virtual database operation

A *database operation* whose source database is the Avaki *virtual database* itself. Use virtual database operations if you want to encapsulate and reuse SQL SELECT queries against *SQL views* (provisioned or generated).

web services client

See *WS client*.

WS client

(Also web services client.) A tool or a piece of code that is part of a customer application and that makes SOAP calls to web services on an Avaki grid server. The SOAP calls can request data from the Avaki *data catalog*, from a *database operation*, or from a *data service*.

Master Index

In electronic copies of this book, the index links to other books in the documentation set work only as long as the PDF files are stored in the same directory.

Key

AD: *Administration Guide*
API: *API Guide*
C: *Command Reference*
O: *Overture*
P: *Provisioning & Advanced Data Integration Guide*
S: *Data Integration with Avaki Studio*

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X