Release Bulletin Sybase® Avaki® EII 7.1 (Data Federation)

Document ID: DC00615-01-0710-02
Last revised: February 27, 2007

Note This release bulletin may be updated after the product is released. To see the latest version, go to http://infocenter.sybase.com, click the Avaki EII 7.1 link in the left pane, and select the Avaki 7.1 release bulletin.

Supported platforms

Sybase® Avaki® EII 7.1.0 supports the following platforms:

Avaki servers (all types) and command clients	• Intel/Windows XP Professional SP2 • Intel/Windows 2003 Server Enterprise Edition SP1 • Intel/Windows 2003 Server Standard Edition SP1 • Intel/Red Hat Enterprise Linux ES 4.0 • Intel/SuSE Linux 9.2 • IBM AIX 5.3 • SPARC/Solaris 9 or 10 • HP-UX 11.11 PA-RISC • HP-UX 11.23 Itanium
Avaki Studio	• Intel/Windows XP Professional SP2 • Intel/Windows 2003 Server Enterprise Edition SP1 • Intel/Windows 2003 Server Standard Edition SP1

Avaki now part of Data Integration Suite; available stand-alone for HP only

Avaki EII 7.1 is available as a stand-alone product only for HP PA-RISC and Itanium. Likewise, the stand-alone version of Avaki Studio is available only to customers who buy Avaki for HP. For all other platforms listed above, Sybase delivers Avaki EII 7.1 in the form of the Data Federation component of Data Integration (DI) Suite 1.1; Avaki Studio is delivered as Data Federation Studio, part of the Workspace component of the DI Suite.

Note Sybase is changing the name of its EII offering from *Avaki* to *Data Federation*. Both names appear in the user interfaces and in the documentation, but they refer to the same product.

Minimum memory requirements

The minimum RAM recommended to run Avaki/Data Federation server and client components has been increased. These recommendations apply to all platforms. For grid servers or GDCs with large data processing loads (heavy database activity, many web services, etc.), consider increasing memory above these minimums:

- Grid domain controllers: 1GB
- Grid servers: 1GB
- Share servers: 1GB
- Data grid access servers: 1GB
- Proxy servers: 1GB
- Command clients: 256MB
- Avaki Studio: 1GB

Note These specifications are for memory in addition to that used by the OS and any other processes running on the machine.

New capabilities

This release includes the following new features:

- SQL engine enhancements
- SySAM license management
- BLOB and CLOB support in SQL views
- JRE upgrade
- Internal database upgrade
  Avaki EII 7.1.0 includes an upgrade to Avaki's internal database that requires migration to a new file format. To prepare your domain for the migration, you'll perform a few preliminary steps described in the release upgrade procedure. The database migration occurs automatically when you restart your grid servers at the end of the upgrade process.

- Internationalization improvements
  Avaki EII 7.1.0 includes improvements to multibyte character support (see "Internationalization issues" on page 23 for details). These improvements do not constitute full internationalization of Avaki EII.

New features in this release are documented in full in *New Features in Sybase Avaki EII 7.1 (Data Federation)*. To see this document, go to http://infocenter.sybase.com, click the Avaki EII 7.1 link in the left pane, and select the new features title from the list.

Special installation instructions

In this section:

Where to find installation instructions

For HP platforms, start with the special HP procedures in this section: "Modifying kernel variables on HP machines," below, and "Mounting installation CDs on HP machines" on page 6. Then locate and follow the installation instructions listed below.

For all platforms, use the following installation and upgrade documentation:

1. Follow either the Data Integration Suite installation instructions (in the DI Suite installation guide) or the Avaki installation instructions (in the 7.0 version of the *Sybase Avaki EII Administration Guide*).
2. Follow the set-up instructions for SySAM license management in *New Features in Sybase Avaki EII 7.1 (Data Federation)*. To see this document, go to http://infocenter.sybase.com, click the Avaki EII 7.1 link in the left pane, and select the new features title from the list.
3. If you're upgrading to this release, go on to "Upgrading to Avaki 7.1" on page 8 when you complete the instructions in the *Sybase Avaki EII Administration Guide*.

Modifying kernel variables on HP machines

To provide sufficient memory for threads spawned by Avaki servers running on HP-UX systems, you must change the values of certain kernel variables. Make these changes before you install Avaki EII software for the first time. Make them on every machine that will host a server in an Avaki domain.

To change the values of kernel variables, follow these steps:

Step 1 Run **sam**, the system administrator configuration tool, and use it to change the following kernel values:

Variable name	Formula	Default value	New value
maxusers	—	32	256
nproc	$20 + (8 \times maxusers)$	276	2068
nkthread	$\frac{nproc \times 7}{4} + 16$	499	3635
max_thread_proc	$< nkthread$	64	1024
ncallout	$16 + nproc$	292	2084
maxfiles	—	60	256

Note Most of the values recommended in the table are based on a *maxusers* value of 256, which Sybase considers a good value for a typical Avaki server. If you need to use a different *maxusers* value, use the formulas in the table to calculate values for the other variables, or contact Sybase technical support for assistance. The values in the "New value" column are minimums—we don't recommend using values lower than these.

Step 2 Reboot the machine to make the changes take effect.

When you finish this procedure, go on to "Mounting installation CDs on HP machines," below.

Mounting installation CDs on HP machines

Certain files on the installation CDs do not show up correctly unless you mount the CDs as shown here. These instructions apply to both first-time installations and upgrades on HP-UX PA-RISC and Itanium.

Step 1 If you're running HP-UX 11.11 (on PA-RISC), make sure you have the following patches (or patches superseding these) installed:

- PHCO_25841
- PHKL_26269
- PHKL_28025

If you don't have these patches installed, use the **pfs_mount** command (rather than **mount**) to mount the installation CD.

Step 2 Use a **mount** command of this form to mount the installation CD:

```
mount -F cdfs -o ro,rr <CDROM device> /cdrom
```

The "rr" refers to the Rockridge format option.

Proceeding with the installation

When you've mounted the CD, proceed as follows:

1. Follow the Avaki installation instructions for Unix platforms in the 7.0 version of the *Sybase Avaki EII Administration Guide*.
2. Follow the set-up instructions for SySAM license management in *New Features in Sybase Avaki EII 7.1 (Data Federation)*. To see this document, go to http://infocenter.sybase.com, click the Avaki EII 7.1 link in the left pane, and select the new features title from the list.
3. If you're upgrading to this release, go on to "Upgrading to Avaki 7.1" on page 8 when you complete the instructions in the *Sybase Avaki EII Administration Guide*.

Interoperability and upgrade path

Avaki 7.1 (for HP) and Data Integration Suite 1.1 (for all other platforms) is intended for Avaki servers of all types (grid servers including GDCs, share servers, data grid access servers, and proxy servers), command line clients, and Avaki Studio/Data Federation Studio instances. Table 1 on page 8 shows the upgrade paths from various previous releases to this new release.

Upgrade all the Avaki servers, command line clients, and Avaki Studio/Data Federation Studio instances in an Avaki domain at the same time. Interoperation of components (servers, command line clients, Studio) running different versions of Avaki software has not been tested.

Use Avaki 7.0 documentation for Avaki 7.1

For Avaki 7.1 and Data Integration Suite 1.1 Data Federation, use the Avaki 7.0 documentation set, plus *New Features in Sybase Avaki EII 7.1 (Data Federation)*. Avaki documentation can be found in the docs subdirectory of your Avaki/Data Federation installation directory, and also at http://infocenter.sybase.com.

JRE upgrade alternative

If you choose not to upgrade to Avaki 7.1.0, or if you don't upgrade right away, Sybase recommends that you upgrade the JREs of the servers in your Avaki 6.2.1 or Avaki 7.0 domain. The JRE upgrade fixes an RSA security problem and addresses the 2007 changes in Daylight Saving Time in the US and Canada.

The JRE upgrade procedure for Avaki 6.2.1 and 7.0 is available at http://infocenter.sybase.com—click the Avaki EII 7.0 link in the left pane, then select the Avaki 6.2/7.0 release bulletin.

TABLE 1. Upgrade path

If you're running...	Upgrade to...	...which is available for download at
Avaki 6.2.1 on HP-UX	Avaki 7.1	Sybase Product Download Center (SPDC): https://sybase.subscribenet.com
Avaki 6.2.1 on any non-HP platform	Avaki 7.0, then to Data Integration Suite 1.1 Data Federation **Note**: Avaki servers running Avaki 6.2.1 must be upgraded to Avaki 7.0 before they can be upgraded to Data Integration Suite 1.1. For instructions on upgrading from Avaki 6.2.1 to Avaki 7.0, see the *Sybase Avaki EII Administration Guide*.	
Avaki 7.0 on any non-HP platform	Data Integration Suite 1.1 Data Federation	
Data Integration Suite 1.0 Data Federation	Data Integration Suite 1.1 Data Federation	

Note Avaki Studio is now distributed as an integrated component of Sybase WorkSpace, which is available by itself or as part of the Data Integration Suite. Within WorkSpace and the DI Suite, Avaki Studio is called *Data Federation Studio*. Avaki Studio is still delivered as a stand-alone component for HP platforms.

Upgrading to Avaki 7.1

This section explains how to upgrade an Avaki domain from Avaki version 6.2.1 (for HP platforms only) or Avaki version 7.0 (all other platforms) to version 7.1.0; the same procedures are used to upgrade from Avaki 7.0 to Data Integration Suite 1.1 Data Federation, or to upgrade Data Federation from Data Integration Suite 1.0 to Data Integration Suite 1.1. Sybase recommends that the upgrade procedures be performed by or with the assistance of a member of the Sybase Professional Services group.

If you're upgrading a non-HP machine from a release earlier than 7.0.0, you must upgrade to the highest available EBF (patch version) of Avaki 7.0.0 before you upgrade to 7.1.0. For instructions on upgrading to 7.0, see the 7.0 version of the *Sybase Avaki EII Administration Guide*. For the latest EBF, go to http://downloads.sybase.com. Log in and select Avaki EII.

In this section:

Before you upgrade

If you're not sure which software version is running on your Avaki servers, use the **avaki upgrade --info** command to display that information:

avaki upgrade --info --all displays version information for all grid servers and share servers in the Avaki domain.

avaki upgrade --info --server=<server-name> displays version information for the specified grid server or share server.

In the procedures in this release bulletin,

`$AVAKI_OLD` refers to the directory on each machine where the older version of Avaki/Data Federation software is installed. (This might be C:\AvakiDataGrid70, for example, or C:\AvakiDataGrid62 for an HP installation.)

`$AVAKI_NEW` refers to the directory where you install the newer version of Avaki/Data Federation software. (This might be C:\SYBASE\DF-7_1—but don't create it yet.)

These must be *two separate* directories.

Planning your upgrade project

Treat the upgrade of your Avaki domain as a formal project and understand that its success depends on the time, resources, and planning you devote to it. We strongly suggest that you prepare a written plan before starting the upgrade. Your upgrade plan should include:

- **Enough time to complete the upgrade process and test the upgraded domain.** The time required depends on the number, type, and configuration of the servers in your domain. Here are some rules of thumb on how much time to allow for each server:

Server type	Time to upgrade	Notes
Grid server including GDC	1 to 1.5 hours	Upgrade time varies based on the number of Avaki shares, database connectors, database operations, data services, and view generators on each grid server.
DGAS	30 minutes	
Share server, proxy server, and Avaki Studio	20 minutes	

- **An execution plan** that includes checklists of tasks for each Avaki server and general tasks. You can use the procedures in this release bulletin ("Preparing your domain for the upgrade," below, "Copying internal data: non-HP platforms" on page 12 or "Copying internal data: HP platforms" on page 16, and "Starting the upgraded servers" on page 20) as starting points for the checklists.

- **A test plan** that ensures that the upgraded domain is fit to return to service. See "Testing" on page 22.

  Your test plan should include checks on a frequently run scheduled task on each grid server that hosts scheduled tasks. If your domain doesn't have any scheduled tasks that run frequently, create a scheduled task that runs at 5-minute intervals so you can use it to verify that scheduled tasks are working after the upgrade.

Preparing your domain for the upgrade

Follow these steps to prepare your Avaki domain for the upgrade:

Step 1 Examine the log of each Avaki server in the domain. If you notice any problems (such as servers failing to communicate with one another), resolve them before continuing with the upgrade. For instructions on finding the log for a server, see the *Sybase Avaki EII Administration Guide*.

Step 2 On every machine that hosts an Avaki server, an Avaki Studio instance, or a command client, install the new Avaki software version, taking care to ensure that *the old and new installations are in different directories*.

Caution Do not start any 7.1.0 servers until you are directed to do so in a later procedure. If you start servers too soon, the data in your installation directories may be compromised. To recover from a premature start, re-install the new software.

Step 3 Copy any database drivers in the old <Avaki-install-dir>/drivers directory to the same location in the 7.1.0 installation:

```
$AVAKI_OLD/drivers/* to
$AVAKI_NEW/drivers/*
```

Step 4 Terminate all grid activities by all users. For example, make sure all users are logged out from their web user interfaces, that all instances of Avaki Studio are disconnected from their servers, and that no user is accessing any file or directory on the grid from an NFS or CIFS DGAS client.

Step 5 Unmount all the NFS clients of every DGAS in the Avaki domain.

Step 6 Shut down all Avaki servers from the older installation in this order:

- DGASes and proxy servers (in any order)
- Share servers (these must all be shut down before you shut down any grid server)
- Ordinary grid servers
- Secondary GDC, if any
- Primary GDC

Refer to the 7.0 version of the *Sybase Avaki EII Administration Guide* for details on how to shut down each kind of server.

Caution Do not restart any servers (in either the old installation or the new one) until you have completed all the upgrade procedures. If you restart servers too soon, the data in your installation directories may be compromised. To recover from a premature start, return to Step 2 of this procedure and perform the upgrade again.

Step 7 Unregister any Avaki servers that are registered as services (for auto-restart).

Step 8 Using the copy tools provided by your operating system, make back-up copies of the old Avaki installation directories of every machine in the Avaki domain.

Go on to one of the following sections:

- "Copying internal data: non-HP platforms," below
- "Copying internal data: HP platforms" on page 16

Copying internal data: non-HP platforms

Use the procedures in this section only as part of an upgrade of non-HP machines from Avaki 7.0 or Data Integration Suite 1.0 to Data Integration Suite 1.1. For instructions on copying internal data on HP machines, see "Copying internal data: HP platforms" on page 16.

In this section:

- "Copying internal data for grid servers/GDCs," below
- "Copying internal data for share servers, proxy servers, DGASes, and Studio" on page 14

Copying internal data for grid servers/GDCs

Follow these steps to copy internal databases, caches, and bindings files to the new installation and to configure system properties. Perform these steps on every grid server in your Avaki domain, including the GDC.

Step 1 On each grid server including the GDC, create a new upgrade directory in the 7.1.0 installation as follows:

```
$AVAKI_NEW/jboss/server/grid-server/upgrade
```

Step 2 Copy the db directory from the 7.0 installation to the upgrade directory you just created in the 7.1.0 installation:

```
$AVAKI_OLD/jboss/server/grid-server/db to
$AVAKI_NEW/jboss/server/grid-server/upgrade
```

resulting in

```
$AVAKI_NEW/jboss/server/grid-server/upgrade/db
```

The presence of the .../upgrade/db directory triggers Avaki's database migration facility to convert the directory contents when the grid server is restarted. (Do not restart the grid server until you're directed to do so in a later procedure.) When it finishes migrating the directory, the grid server changes the directory name from db to db-bak,

which prevents the migration process from running again the next time the server restarts. The migrated database files are placed in $AVAKI_NEW/jboss/server/grid-server/db.

Step 3 Copy the contents of the caches directory from the 7.0 installation to the 7.1.0 installation:

```
$AVAKI_OLD/caches/* to
$AVAKI_NEW/caches/*
```

Step 4 If your domain was originally installed with Avaki 6.2 or earlier, copy the bindings.xml files for every grid server, including the GDC, from the 7.0 installation to the 7.1.0 installation. This is necessary because several of the default ports used by grid servers changed in Avaki 7.0, but existing domains must continue to use the ports with which they were originally configured. If in doubt, it's safest to copy each server's bindings.xml file from the old installation to the new one.

```
$AVAKI_OLD/jboss/server/grid-server/conf/bindings.xml to
$AVAKI_NEW/jboss/server/grid-server/conf/bindings.xml
```

Step 5 If during the original deployment any default ports were changed on any grid servers, copy the bindings.xml file for every affected grid server from the 7.0 installation to the 7.1.0 installation. This is necessary because existing domains must continue to use the ports with which they were originally configured. If in doubt, it's safest to copy each server's bindings.xml file from the old installation to the new one.

```
$AVAKI_OLD/jboss/server/grid-server/conf/bindings.xml to
$AVAKI_NEW/jboss/server/grid-server/conf/bindings.xml
```

Step 6 If during the original deployment any default ports were changed for the internal share servers on any grid servers, including the GDC, copy the following files for the affected servers from the 7.0 installation to the 7.1.0 installation:

```
$AVAKI_OLD/jboss/server/grid-server/conf/shareserver.ports to
$AVAKI_NEW/jboss/server/grid-server/conf/shareserver.ports
```

This is necessary because existing domains must continue to use the ports with which they were originally configured. If in doubt, it's safest to copy each server's bindings.xml file from the old installation to the new one.

Step 7 If during the original deployment any system properties files or startup scripts were changed, make equivalent changes in the new grid. For example,

- If `$AVAKI_OLD/jboss/server/grid-server/conf/system.properties` was changed in order to provide a specific DNS name for the machine, make an equivalent change to `$AVAKI_NEW/jboss/server/grid-server/conf/system.properties`. See the *Sybase Avaki EII Administration Guide* for instructions.

Note The fully qualified hostname for each Avaki server in the upgraded data grid must be exactly the same as the hostname for the corresponding server in the old grid.

- If `$AVAKI_OLD/grid-server` was changed to increase the memory limit for the Java virtual machine, make an equivalent change to `$AVAKI_NEW/grid-server`.

Copying internal data for share servers, proxy servers, DGASes, and Studio

Follow these steps to copy internal databases, caches, bindings files, and workspaces to the new installation and to configure system properties.

Step 1 On share servers and proxy servers only, rename the 7.1.0 db directory so it will be available as a back-up if needed:

```
$AVAKI_NEW/jboss/server/share-server/db to
$AVAKI_NEW/jboss/server/share-server/db-bak

$AVAKI_NEW/jboss/server/proxy-server/db to
$AVAKI_NEW/jboss/server/proxy-server/db-bak
```

Step 2 Also on share servers and proxy servers only, copy the db directory from the 7.0 installation to the 7.1.0 installation:

```
$AVAKI_OLD/jboss/server/share-server/db to
$AVAKI_NEW/jboss/server/share-server/db

$AVAKI_OLD/jboss/server/proxy-server/db to
$AVAKI_NEW/jboss/server/proxy-server/db
```

Step 3 On DGASes only, copy the contents of the DGAS/dgas_db directory from the 7.0 installation to the 7.1.0 installation:

```
$AVAKI_OLD/DGAS/dgas_db/* to
$AVAKI_NEW/DGAS/dgas_db/*
```

Step 4 Copy any DGAS caches from the 7.0 installation to the 7.1.0 installation:

```
$AVAKI_OLD/DGAS/cache/* to
$AVAKI_NEW/DGAS/cache/*
```

Step 5 If any Avaki Studio workspace directories are located in the Avaki installation directory, copy each workspace directory from the 7.0 installation to the 7.1.0 installation. The following example assumes the workspace for both the old and the new installations is in the install directory:

```
$AVAKI_OLD/workspace/* to
$AVAKI_NEW/workspace/*
```

Note If you have configured a workspace directory outside the Avaki installation directory (which is the default), you can simply enter its path when you start the new version of Avaki Studio—there's no need to copy the directory.

Step 6 If during the original deployment any default ports were changed on proxy servers or share servers, copy the following files for the affected servers from the 7.0 installation to the 7.1.0 installation:

```
$AVAKI_OLD/jboss/server/proxy-server/conf/bindings.xml to
$AVAKI_NEW/jboss/server/proxy-server/conf/bindings.xml

$AVAKI_OLD/jboss/server/share-server/conf/bindings.xml to
$AVAKI_NEW/jboss/server/share-server/conf/bindings.xml

$AVAKI_OLD/jboss/server/share-server/conf/shareserver.ports to
$AVAKI_NEW/jboss/server/share-server/conf/shareserver.ports
```

This is necessary because existing domains must continue to use the ports with which they were originally configured. If in doubt, it's safest to copy each server's bindings.xml file from the old installation to the new one.

Step 7 If during the original deployment any system properties files or startup scripts were changed, make equivalent changes in the new grid. For example,

- If `$AVAKI_OLD/jboss/server/<server-type>/conf/system.properties` was changed in order to provide a specific DNS name for the machine, make an equivalent change to `$AVAKI_NEW/jboss/server/<server-type>/conf/system.properties`. See the *Sybase Avaki EII Administration Guide* for instructions.

Note The fully qualified hostname for each Avaki server in the upgraded data grid must be exactly the same as the hostname for the corresponding server in the old grid.

- If `$AVAKI_OLD/share-server` was changed to increase the memory limit for the Java virtual machine, make an equivalent change to `$AVAKI_NEW/share-server`.

Proceed to "Starting the upgraded servers" on page 20.

Copying internal data: HP platforms

Use the procedure in this section only on HP platforms as part of an upgrade from Avaki 6.2.1 to Avaki 7.1.0. For instructions on copying internal data on other machines, see "Copying internal data: non-HP platforms" on page 12.

In this section:

Copying internal data for grid servers/GDCs

Follow these steps to copy internal databases, caches, and bindings files to the new installation and to configure system properties. Perform these steps on every grid server in your Avaki domain, including the GDC.

The purpose of copying the 6.2.1 and 7.1.0 db directories back and forth (Step 4 and Step 5) is to create a composite copy of the db directory that contains both of the following:

- your data and server configurations from the 6.2.1 installation, and
- the new metadata from the 7.1.0 installation.

Step 1 On each grid server including the GDC, make a back-up copy of the db directory from the 7.1.0 installation:

```
$AVAKI_NEW/jboss/server/grid-server/db to
$AVAKI_NEW/jboss/server/grid-server/db-71-bak
```

Step 2 Create a new upgrade directory in the 7.1.0 installation:

```
$AVAKI_NEW/jboss/server/grid-server/upgrade
```

Step 3 Copy the 7.1.0 installation's db directory into the upgrade directory:

```
$AVAKI_NEW/jboss/server/grid-server/db to
$AVAKI_NEW/jboss/server/grid-server/upgrade/db
```

Step 4 Copy the 6.2.1 db directories on top of the 7.1.0 upgrade/db directories:

```
$AVAKI_OLD/jboss/server/grid-server/db/* to
$AVAKI_NEW/jboss/server/grid-server/upgrade/db/*
```

Step 5 Copy the 7.1.0 db directories into the upgrade/db directory, overwriting any 6.2.1 files:

```
$AVAKI_NEW/jboss/server/grid-server/db/*
$AVAKI_NEW/jboss/server/grid-server/upgrade/db/*
```

The presence of the .../upgrade/db directory triggers Avaki's database migration facility to convert the directory contents when the grid server is restarted. (Do not restart the grid server until you're directed to do so in a later procedure.) When it finishes migrating the directory, the grid server changes the directory name from db to db-bak, which prevents the migration process from running again the next time the server restarts. The migrated database files are placed in $AVAKI_NEW/jboss/server/grid-server/db.

Step 6 On each grid server including the GDC, copy the contents of the caches directory from the 6.2.1 installation to the 7.1.0 installation:

```
$AVAKI_OLD/caches/* to
$AVAKI_NEW/caches/*
```

Step 7 Copy the bindings.xml files for every grid server, including the GDC, from the 6.2.1 installation to the 7.1.0 installation. This is necessary because several of the default ports used by grid servers changed in Avaki 7.0, but existing domains must continue to use the ports with which they were originally configured. If in doubt, it's safest to copy each server's bindings.xml file from the old installation to the new one.

```
$AVAKI_OLD/jboss/server/grid-server/conf/bindings.xml to
$AVAKI_NEW/jboss/server/grid-server/conf/bindings.xml
```

Step 8 If during the original deployment any default ports were changed on any grid servers, copy the bindings.xml file for every affected grid server from the 6.2.1 installation to the 7.1.0 installation. This is necessary because existing domains must continue to use the ports with which they were originally configured. If in doubt, it's safest to copy each server's bindings.xml file from the old installation to the new one:

```
$AVAKI_OLD/jboss/server/grid-server/conf/bindings.xml to
$AVAKI_NEW/jboss/server/grid-server/conf/bindings.xml
```

Step 9 If during the original deployment any default ports were changed for the internal share servers on any grid servers, including the GDC, copy the following files for the affected servers from the 6.2.1 installation to the 7.1.0 installation:

```
$AVAKI_OLD/jboss/server/grid-server/conf/shareserver.ports to
$AVAKI_NEW/jboss/server/grid-server/conf/shareserver.ports
```

This is necessary because existing domains must continue to use the ports with which they were originally configured. If in doubt, it's safest to copy each server's bindings.xml file from the old installation to the new one.

Step 10 If during the original deployment any system properties files or startup scripts were changed, make equivalent changes in the new grid. For example,

- If `$AVAKI_OLD/jboss/server/grid-server/conf/system.properties` was changed in order to provide a specific DNS name for the machine, make an equivalent change to `$AVAKI_NEW/jboss/server/grid-server/conf/system.properties`. See the *Sybase Avaki EII Administration Guide* for instructions.

Note The fully qualified hostname for each Avaki server in the upgraded data grid must be exactly the same as the hostname for the corresponding server in the old grid.

- If `$AVAKI_OLD/grid-server` was changed to increase the memory limit for the Java virtual machine, make an equivalent change to `$AVAKI_NEW/grid-server`.

Copying internal data for share servers, proxy servers, DGASes, and Studio

Follow these steps to copy internal databases, caches, bindings files, and workspaces to the new installation and to configure system properties.

Step 1 On share servers and proxy servers only, rename the 7.1.0 db directory so it will be available as a back-up if needed:

```
$AVAKI_NEW/jboss/server/share-server/db to
$AVAKI_NEW/jboss/server/share-server/db-bak

$AVAKI_NEW/jboss/server/proxy-server/db to
$AVAKI_NEW/jboss/server/proxy-server/db-bak
```

Step 2 Also on share servers and proxy servers only, copy the db directory from the 6.2.1 installation to the 7.1.0 installation:

```
$AVAKI_OLD/jboss/server/share-server/db to
$AVAKI_NEW/jboss/server/share-server/db

$AVAKI_OLD/jboss/server/proxy-server/db to
$AVAKI_NEW/jboss/server/proxy-server/db
```

Step 3 On DGASes only, copy the contents of the DGAS/dgas_db directory from the 6.2.1 installation to the 7.1.0 installation:

```
$AVAKI_OLD/DGAS/dgas_db/* to
$AVAKI_NEW/DGAS/dgas_db/*
```

Step 4 Copy any DGAS caches from the 6.2.1 installation to the 7.1.0 installation:

```
$AVAKI_OLD/DGAS/cache/* to
$AVAKI_NEW/DGAS/cache/*
```

Step 5 If any Avaki Studio workspace directories are located in the Avaki installation directory, copy each workspace directory from the 6.2.1 installation to the 7.1.0 installation. The following example assumes the workspace for both the old and the new installations is in the install directory:

```
$AVAKI_OLD/workspace/* to
$AVAKI_NEW/workspace/*
```

Note If you have configured a workspace directory outside the Avaki installation directory (which is the default), you can simply enter its path when you start the new version of Avaki Studio—there's no need to copy the directory.

Step 6 If during the original deployment any default ports were changed on proxy servers or share servers, copy the following files for the affected servers from the 6.2.1 installation to the 7.1.0 installation:

```
$AVAKI_OLD/jboss/server/proxy-server/conf/bindings.xml to
$AVAKI_NEW/jboss/server/proxy-server/conf/bindings.xml

$AVAKI_OLD/jboss/server/share-server/conf/bindings.xml to
$AVAKI_NEW/jboss/server/share-server/conf/bindings.xml
```

`$AVAKI_OLD/jboss/server/share-server/conf/shareserver.ports` to `$AVAKI_NEW/jboss/server/share-server/conf/shareserver.ports`

This is necessary because existing domains must continue to use the ports with which they were originally configured. If in doubt, it's safest to copy each server's bindings.xml file from the old installation to the new one.

Step 7 If during the original deployment any system properties files or startup scripts were changed, make equivalent changes in the new grid. For example,

- If `$AVAKI_OLD/jboss/server/<server-type>/conf/system.properties` was changed in order to provide a specific DNS name for the machine, make an equivalent change to `$AVAKI_NEW/jboss/server/<server-type>/conf/system.properties`. See the *Sybase Avaki EII Administration Guide* for instructions.

Note The fully qualified hostname for each Avaki server in the upgraded data grid must be exactly the same as the hostname for the corresponding server in the old grid.

- If `$AVAKI_OLD/share-server` was changed to increase the memory limit for the Java virtual machine, make an equivalent change to `$AVAKI_NEW/share-server`.

Starting the upgraded servers

Step 1 In the 7.1.0 installation (`AVAKI_NEW`) on each grid server including the GDC, use a text editor such as Notepad or Wordpad to open the file DF-7_1/resources/jndi.properties. Look for the following line:

```
java.naming.provider.url=localhost:3099
```

Step 2 Make sure that the number at the end of that line (3099 in the example above) is the connect port number for this grid server or grid domain controller. If your domain was first installed with Avaki 6.2 or earlier using default port numbers, you'll change the port number here from 3099 to 1099.

If you're not sure what the connect port number for your server is, open the bindings.xml file (in DF-7_1/jboss/server/grid-server/conf) and search for "jboss.bind.address." The port bound on that line is the connect port.

Step 3 Start the grid servers from the 7.1.0 installation. Bring up the servers in the following order:

- The primary GDC
- The secondary GDC, if any
- All ordinary grid servers

When you start each server, it performs a migration of its database directory as part of it initialization. This process makes the boot take a few minutes longer than usual.

Step 4 On each grid server (including GDCs), check the server log to confirm that the upgrade worked—the log will include messages that say "State from a 7.0 server now being loaded on a 7.1 server" (if you're upgrading a server on an HP machine, the message says "State from a Kite server..." or "State from a Kestrel server....") and "Upgrade completed successfully." If these messages are not in the log, contact Sybase technical support. (The grid server log is DF-7_1/jboss/server/grid-server/log/server.log.)

Step 5 Shut down and restart all the grid servers (including GDCs). When you restart the servers, follow the same order used in Step 3.

Step 6 Start the share servers, DGASes and proxy servers from the 7.1.0 installation. Do not start any share servers until all the grid servers in the domain are running.

Post-upgrade tasks

After you upgrade an Avaki domain, perform the following tasks.

1. Remount DGAS NFS clients. See the *Sybase Avaki EII Administration Guide* for instructions.
2. Reconnect command clients (see the *Sybase Avaki EII Administration Guide*) and Avaki/Data Federation Studio (see *Data Integration with Sybase Avaki Studio*).
3. Test the upgraded domain (see "Testing," below).
4. Resume grid activity. You can do this immediately, though you might choose to wait until the testing phase is complete.

Testing

Follow your test plan to ensure that your upgraded Avaki domain is functioning properly. Your test plan should include the following:

- Look in the server logs for indications of the success or failure of the upgrade.
- Have multiple users (including several nonadministrative users and users from each authentication service) log in to the domain.
- Make sure that scheduled tasks (such as data service executions and share rehashes) are running, including any schedules you created to test the upgrade.
- Make sure you can create new schedules.
- Make sure you can access data in Avaki shares.
- Make sure remote and local caches are accessible and are being populated. (See the *Sybase Avaki EII Provisioning and Advanced Data Integration Guide* for instructions on viewing caches and managing cache services.)

Defects fixed in this release

Release 7.1.0

Internationalization issues

Description	CR Number
Table names using multibyte character sets do not function in Avaki Studio	440028
Unable to create a data service using a database connector whose name includes multibyte Korean characters	444142
Unable to create a user with multibyte Korean characters in e-mail address	444144
Schema browser in Avaki web UI does not show Korean characters correctly	444146
Unable to create a schedule exclusion whose name includes Korean characters	444276
Unable to create a virtual SQL view using a database connector whose name includes Korean characters	444277
Unable to browse into an Avaki share whose name includes Korean characters	445372
Unable to move a directory whose name includes Korean characters	445417
Unable create a generated view in a directory whose name includes Korean characters	445418
The Move Directory screen in the web UI does not display Korean characters in directory names	445423
The Link Object screen in the web UI does not display Korean characters in object names	445425

Expand/collapse problem in Avaki Studio/Data Federation Studio

Description	CR Number
Unable to collapse and expand schema in view model editor as you could in previous releases	438156
Problems importing, displaying view models created in Avaki 6.2.1	442401

Known problems

Release 7.1.0

Installer/internationalization issues

Description/Workaround	CR Number
The Avaki installer fails on Chinese Windows XP when the login name contains multi-byte characters. The installer process disappears after displaying its initial pop-up dialog. *Workaround*: Do either of the following: • Create an account whose name uses no multibyte characters and use it to install Avaki. • Install Avaki on a system running English Windows and a Chinese language pack.	443863

SQL-related issues

Description/Workaround	CR Number
Avaki does not support the SQL Anywhere data type *oldbit*. Attempting to use the data type *oldbit* causes an execution error.	440369

Description/Workaround	CR Number
An out of memory error occurs when a CLOB or BLOB database file is larger than the Avaki grid server memory. If you attempt to get data from a database table that contains BLOB or CLOB data, and the CLOB/BLOB data includes a file larger than the Avaki server's memory size, an out of memory error occurs. *Workaround*: Increase the Avaki grid server's heap size. You can change this memory allocation by editing the grid-server.bat file in the top-level installation directory; change the Xmx value to the desired setting. The line that you should edit is similar to this: `if "%SERVER_TYPE%" == "grid-server" set VM_ARGS=-server -Xms256m -Xmx512m -Dsun.rmi.transport.connectionTimeout=120000 -Xrs`	445892 & 449064
Query engine allows (and executes) invalid SQL with GROUP BY. For example, if you use this SQL statement: `select c1_uid, count(c6_time) as count_c6, count(c7_timestamp) as count_c7 from functions_test group by c3_char` an error should be returned stating that c1_uid must appear in the GROUP BY clause or be removed from SELECT list. Instead, Avaki executes the query and provides incorrect results.	445970
MAX returns inconsistent data from database, provisioned SQL view, database operation SQL view, and data service SQL view. For example, `select MAX(col_name)` should return the same value in all cases, but the results from the database and a provisioned SQL view are inconsistent with those of a database operation SQL view and a data service SQL view.	450227
If many virtual objects exist, the first query of a virtual database requires several minutes. After the server comes up, the first virtual database query might require several minutes to give a response. The system might appear to have hung, but the response does eventually appear. Subsequent queries perform as expected.	453290

Description/Workaround	CR Number

A DB2 DATEPART query fails for a provisioned SQL view, but succeeds for a database operation or data service SQL view.
 	
The same DB2 DATEPART query succeeds for a database operation or data service SQL view but fails for a provisioned SQL view.
 	
Workaround: Include a column from the table in the select clause. For example, instead of:
 	
`SELECT DATEPART(year, '2006') from EMP_sv`
 	
Include a column from the table in the query, such as:
 	
`SELECT hiredate, DATEPART(year, '2006') from EMP_sv`
	454478

The DATEPART() error message is unclear when a parameter is missing. If you have a DATEPART() query and leave out a parameter (with or without a comma), the error message does not say that a parameter was missing. For example: `Error: com.avaki.core.util.LoggableException: Error executing database operation: Function DATEPART(Week) isn't valid.` *Workaround*: Make sure you are using the correct syntax for the function that you want to use.	454502
In a DATEPART() query, spaces within parameter values are accepted but give unpredictable results. DATEPART() takes two parameters. If you include any spaces in either parameter value, you can get unpredictable results. Spaces should cause an error message. *Workaround*: Verify that no spaces exist within parameter values.	454613

Description/Workaround	CR Number

DATEDIFF() — For some back-end databases, you might get an error if a column from the table is not included in the SELECT.
 	
For example, the following queries fail:
 	
`avaki virtualdatabase --execute "select DATEDIFF(Day,'2006/11/21', '2009/02/05') from emp_mysql_sv"`
 	
`avaki virtualdatabase --execute "select DATEDIFF(Week,'2006/11/21', '2009/02/05') from emp_mysql_sv"`
 	
Workaround: Include a column from the table in the SELECT clause.
 	
For example, the following queries succeed:
 	
`avaki virtualdatabase --execute "select hiredate,DATEDIFF(Day,'2006/11/21', '2009/02/05') from emp_mysql_sv"`
 	
`avaki virtualdatabase --execute "select hiredate,DATEDIFF(Week,'2006/11/21', '2009/02/05') from emp_mysql_sv"`
	454805

DATEDIFF() produces an unclear error message when the datepart is unsupported. Unsupported dateparts for DATEDIFF are Dayofyear, dy, Weekday, and dw.	455015

In rare circumstances, queries using the HAVING clause can fail.
 	
If you use a HAVING clause in a query (typically when the WHERE clause could have been used instead), the query might fail.
 	
For example, the following two queries succeed for ASE but fail for Oracle:
 	
`avaki virtualdatabase --execute "select * from EMP_ase_sv group by MGR HAVING MGR > 7698"`
 	
`Error: com.avaki.core.util.LoggableException: Error executing database operation:  Field * not found.`
 	
`avaki virtualdatabase --execute "select MGR from EMP_ase_sv group by MGR HAVING MGR > 7698"`
 	
`Error: com.avaki.core.util.LoggableException: Error executing database operation:  Field (EMP_ASE_SV).MGR not found.`
 	
In more complex queries where the HAVING clause must be used, the queries succeed.
	455367

Description/Workaround	CR Number
Queries that use FULL OUTER JOIN can result in errors if the back-end database doesn't support that syntax. The FULL OUTER JOIN portion of a SQL query is always pushed down to the back-end database when only one database connection is involved in the query. Errors are returned if the back-end database does not support FULL OUTER JOIN syntax. Sybase ASE does not support this syntax. However, if a query refers to SQL views based on different database connectors, the Avaki query engine parses the query and it succeeds, even if ASE is involved. *Workaround*: In a FULL OUTER JOIN query involving Sybase ASE or another DBMS that doesn't support FULL OUTER JOIN, include SQL views based on two or more different database connectors.	455528
A virtual database operation with a parameter fails if the parameter's data type is any number type (integer, number, real, decimal). The operation succeeds if the type is any character type (char, varchar).	455706
On AIX systems, additional information appears in the server display window. AIX systems display additional text on the display window; this text does not display for other OSs and is not written to the server log. This informational text is non-vital and has no effect on server operation.	455803

In some cases, a provisioned SQL view using a WHERE clause fails when a database operation or data service SQL view succeeds; ad hoc queries to the table succeed.
 	
The following example query fails:
 	
`avaki virtualdatabase --execute "SELECT HIREDATE, GETDATE() FROM EMP_asa_sv where HIREDATE > DATEADD(year, -24, GETDATE())"`
 	
The following example query succeeds:
 	
`avaki virtualdatabase --execute "SELECT HIREDATE, GETDATE() FROM ASA_selectEMP_ds_sv where HIREDATE > DATEADD(year, -24, GETDATE())"`
 	
Workaround: Use a HAVING clause instead of a WHERE clause.
	455853

Description/Workaround	CR Number
Error messages displayed on the client for some queries can be misleading. When a query results in an error, the error message might be misleading. For example, you might see an error message on the client that contains the following text: `Unknown exception has been thrown from server to client. You may need to add necessary jars to your client classpath.` Check the server log for an accurate description of the error; the server log is located at <install_dir>\jboss\server\grid-server\log.	455855

The DATEADD function can give different results depending on the server platform.
 	
The following examples show queries and results for different server platforms:
 	
`avaki virtualdatabase --execute "select DATEADD(year,-1, '2009') from EMP_ase_sv"`
 	
Solaris and Windows 2003 Server SE 32 bit:
 	
`2008-01-01 00:00:00.0`
 	
AIX:
 	
`2007-12-31 23:00:00.0`
 	
HP:
 	
`2007-12-31 22:00:00.0`
 	
`avaki dbconn ASE15avaki --execute "select DATEADD(year,1, '2006') from EMP"`
 	
Solaris and Windows 2003 Server SE 32 bit:
 	
`2007-01-01 00:00:00.0`
 	
AIX:
 	
`2006-12-31 23:00:00.0`
 	
HP:
 	
`2006-12-31 22:00:00.0`
	455856

Description/Workaround	CR Number
The SUM and AVG(DISTINCT SAL) functions can give inconsistent values depending on whether you are performing them using the JDBC driver or the command line. Some queries that have a numerical result might either round the value to a whole number, or display the exact value; the same query can result in a truncated whole number result when using the JDBC driver. For example, a result could be displayed as 491, 490.5, or 490.5000000; the corresponding result from a provisioned SQL view queried using the Avaki JDBC driver is truncated to 490.	455963

Web service issues

Description/Workaround	CR Number
A web service cannot be used as a data service input. If a web service is used as a data service input, the web service fails, returning 0 as the conversion rate in all cases.	417747
Web service invocation can fail silently during Transform phase. No information is returned when an error is encountered running a web service during the Transform phase of creating an input source. Avaki Studio stays on the same dialog as if nothing has happened.	425944

Web UI issue

Description/Workaround	CR Number
Hyperlinking to files in the Avaki grid fails if the user is not logged in. If you set up a hyperlink to a file in the grid, the link works if the user is logged in to the web UI. If the user is not logged in, the web UI requires that the user log in, but then fails to take the user to the object.	443190

File format issue

Description/Workaround	CR Number
There are ^M characters in the properties files in the resources directory. On Unix platforms, the properties files in the resources subdirectory of the Avaki/Data Federation install directory (DF-7_1/resources) have ^M characters at the end of each line. (If you're upgrading to Avaki 7.1/Data Federation 1.1, you might notice this when you edit the jndi.properties file.) The ^Ms are a cosmetic problem—you can ignore them.	455380

Written by Emily Goodin and Beth Thoenen

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