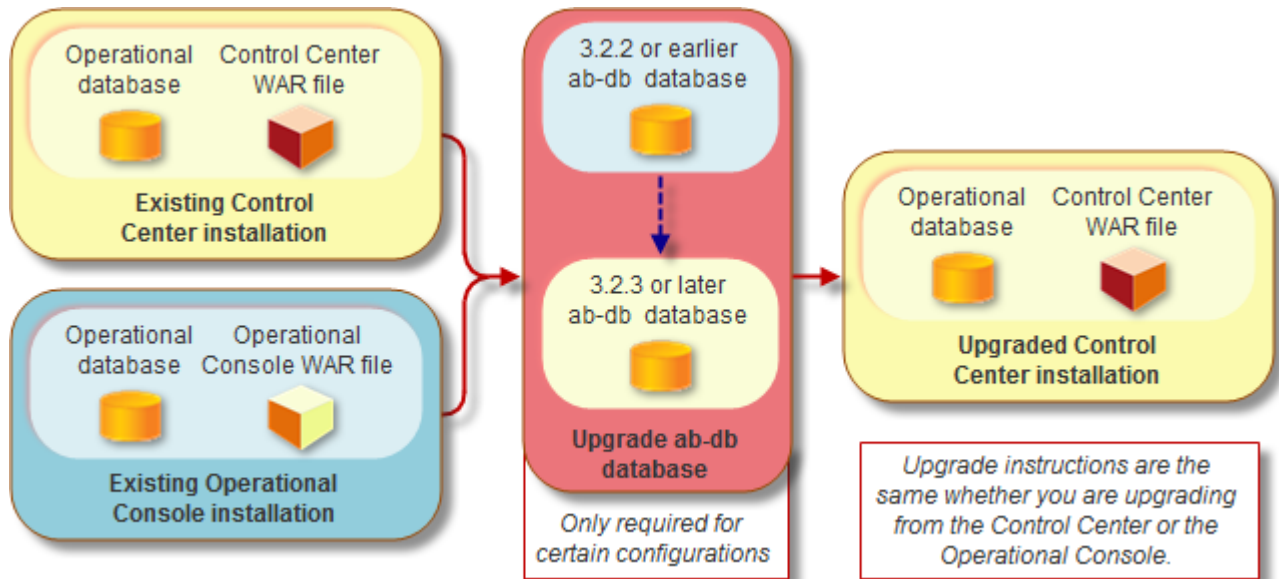


Overview of the Control Center upgrade process



The **Control Center** upgrade procedure replaces an older version of the **Control Center** or Operational Console WAR file and **operational database** schema with a new **Control Center** WAR file and schema while preserving the **operational database** records from the earlier installation. The preserved database records include all configuration settings, user account data and security settings, job and schedule definitions, job histories, and details about monitored hosts, products, and services.

Note the following important considerations about upgrading the **operational database**:

- **Backing up your existing operational database** — Before performing a **Control Center** upgrade, we recommend that you back up your existing **operational database**. For instructions, see ["Backing up your operational database"](#).
- **Upgrading on Co>Operating System 3.2.3 or later** — If your existing **operational database** was created using an **ab-db** version from **Co>Operating System** Version 3.2.2 or earlier, and if your **Co>Operating System** was subsequently upgraded to Version 3.2.3 or later, then you need to upgrade your **operational database** to make it compatible with the 3.2.3 or later version of **ab-db** server software. For instructions, see ["Upgrading an ab-db operational database on Co>Operating System 3.2.3 or later"](#).
- **Upgrading to Control Center Version 3.2.2 or later** — Because of significant database schema changes that were introduced in **Control Center** Version 3.2.2, you may want to purge older records from your pre-3.2.2 **operational database**, if applicable, to prevent an unusually lengthy upgrade process. If you are upgrading from a **Control Center** version prior to 3.2.2 or from any version of the Operational Console, we strongly recommend that you review the information in ["Upgrading the operational database from versions prior to 3.2.2"](#) before you run the **Control Center** `upgrade_database.ksh` script.

This section contains the following topics:

- [Upgrading from the Operational Console to the Control Center](#)
- [Backing up your operational database](#)
- [Upgrading an ab-db operational database on Co>Operating System 3.2.3 or later](#)
- [Upgrading the operational database from versions prior to 3.2.2](#)
- [Adjusting abinitiorc and apphubrc settings](#)

Upgrading from the Operational Console to the [Control Center](#)

Starting with [Control Center](#) Version 3.2.1, [Control Center](#) software replaces Operational Console software Versions 3.1.5 and earlier. [Control Center](#) software extends the job monitoring and scheduling features of the Operational Console, and adds new host, system, product, and service monitoring and management tools. With the exception of occasional patch releases, [Ab Initio](#) has discontinued the Operational Console product line as of Version 3.1.5.x.

Operational Console upgrade summary

If you are currently using Operational Console Version 3.1.5.x or earlier, the procedure for upgrading from the Operational Console to the [Control Center](#) is the same as the previous procedure for upgrading from one Operational Console version to the next.

Note the following:

- When you upgrade an existing Operational Console installation to a new [Control Center](#) installation, your existing Operational Console WAR file is replaced by a new [Control Center](#) WAR file, and your existing [operational database](#) is updated to use the latest [Control Center](#) database schema.
- You can continue to use your existing Operational Console [operational database](#) with the [Control Center](#). All [operational database](#) records from the Operational Console installation are preserved, and no Operational Console operational data is discarded or altered by the [Control Center upgrade_database](#) script.
- The [Control Center operational database](#) schema extends the Operational Console schema, and includes a number of new tables and columns to accommodate [Control Center](#) features that did not exist in the Operational Console. For complete details about the [Control Center operational database](#), see the [Schema Reference](#).

[Control Center](#) and Operational Console feature comparison

The original Operational Console job and scheduling features are a subset of the new [Control Center](#) features that enable you to monitor and manage your overall operational environment, including [Co>Operating System](#) hosts, and [Ab Initio](#) and third-party products and services.

Listed below are summaries of the major [Control Center](#) features that expand upon or replace features in the original Operational Console product.

- **New user interface areas and tools** — To accommodate the new [Control Center](#) features, the [Control Center](#) user interface has been reorganized and streamlined into several new areas:

Area	Description
Home area	Provides a customizable overview of all aspects of your operational infrastructure, and includes a number of new ways to group and visualize information.
Hosts area	Provides monitoring and management tools for Co>Operating System hosts and filesystems.
Products area	Provides monitoring and management tools for Ab Initio and third-party products and services.
Jobs area	Provides the former Operational Console job and job scheduling tools. The old Operational Console Home area is now part of the new Control Center Jobs area.
Status Details for Control Center	Provides centralized monitoring and management tools for Control Center reporters and bridges. You can stop, start, and configure version 3.1.2.x and later reporters directly from the Control Center . Previously, you had to perform those tasks from an operating system command line. Note that bridge configuration tools were previously in the Operational Console Configuration area.
Administrative Tasks menu	Provides a centralized location to go to for many Control Center administrative tasks. This menu includes tools that were previously spread across several Operational Console Configuration areas.

- **Host and filesystem monitoring** — The [Control Center](#) provides host and filesystem monitoring, including information about host status, host metrics, metric history, and metric threshold issues.
- **Product and service monitoring** — The [Control Center](#) provides monitoring and management tools for [Ab Initio](#) and third-party products and services, including status, issues, starting and stopping, editing configuration files, and viewing log files.
- **Predictive job monitoring** — The [Control Center](#) can predict and warn you about potential job delays based on historical job execution times, job dependencies, and job timer data. Predictive job monitoring becomes more accurate as job histories are accumulated.
- **Software key administration** — With the appropriate [Control Center](#) role, you can use the [Control Center](#) to update and return computer keys, and administer user keys and key servers. For the administration of user keys and key servers, the [Control Center](#) replaces the deprecated Key Server Administration Console for a number of tasks, including changing the key server password, installing key bundles, and generating reports.
- **Automated Purge** — The [Control Center](#) enables you to automatically purge records from the [operational database](#) at regularly scheduled times. This automated purging process does not require user intervention, ensures database integrity by intelligently deleting all related records for purged objects, and can purge job, host, product, queue, and reporter histories.

Backing up your operational database

Before proceeding with a [Control Center](#) upgrade, we recommend that you back up your existing [operational database](#). In most cases, you will not need this backup copy, and performing this backup is an optional safety precaution.

Instructions for backing up your [operational database](#) vary by database server:

- If you use an Oracle, Microsoft SQL Server, PostgreSQL, or IBM DB2 database server, see the documentation that is included with those products for backup and restore instructions.
- If you use **ab-db**, use the following procedures to perform a backup and, if necessary, a restore.
NOTE: If you are upgrading an **ab-db** [operational database](#) on [Co>Operating System](#) Version 3.2.3 or later, skip the procedures in this section, and instead proceed with "[Upgrading an ab-db operational database on Co>Operating System 3.2.3 or later](#)".

Backing up an ab-db operational database

► **To back up an ab-db operational database:**

1. Using an [Ab Initio](#) KSH shell on the [Co>Operating System](#) host on which the original **ab-db** instance for the [operational database](#) was created, ensure the following:
 - a. The **ab-db** [operational database](#) instance that you want to back up is running; for example:

```
ab-db status db_name
```

where *db_name* is the name of the original [operational database](#).

- b. The AB_HOME configuration variable points to the [Co>Operating System](#) on which the original **ab-db** instance was created.
2. Enter the following commands to set the required environment variables for the backup:

```
export PATH=$AB_HOME/lib/postgresql/bin:$PATH
export LD_LIBRARY_PATH=$AB_HOME/lib/postgresql/lib:$LD_LIBRARY_PATH
export PGUSER=opdb
export PGPASSWORD=abinitio
export PGPORT=6464
export PGDATABASE=OPDB
```

NOTE: These commands assume the default [operational database](#) name, username, and password that are used in [Control Center](#) and Operational Console standard installations. If your [operational database](#) uses different values, use those values instead of these defaults.

3. Enter the following command to perform the [operational database](#) backup:

```
pg_dump OPDB --file=/old_path/dumpfile --format=tar --verbose
```

where:

- *old_path* — The filesystem path on the original machine where you want to store the database backup file.
- *dumpfile* — The name of the database backup file that you want to create.

Restoring an ab-db operational database

In most cases, your **Control Center** upgrade will proceed without incident, and you will not need to restore the **ab-db operational database** from the backup file that you created in the preceding procedure. However, if you do need to restore your **ab-db operational database** from a backup, use the following procedure.

► **To restore an ab-db operational database from backup:**

1. Using an **Ab Initio** KSH shell on the **Co>Operating System** host on which you want to restore the **ab-db operational database**, ensure the following:

- a. The new **ab-db** instance to which you want to restore **operational database** is running; for example:

```
ab-db status db_name
```

where *db_name* is the name of the **ab-db** instance to which you want to restore the **operational database**.

- b. The **AB_HOME** configuration variable points to the **Co>Operating System** on which the new **ab-db** instance was created.

2. Stop the **Control Center** instance; for example:

```
ab-app stop controlcenter
```

3. Copy the database backup file that you created in Step 3 of "**Backing up an ab-db operational database**" to the location of your choice on the **Co>Operating System** host on which the new **ab-db** instance is running.

4. Enter the following commands to set the required environment variables for the restore:

```
export PATH=$AB_HOME/lib/postgresql/bin:$PATH
export LD_LIBRARY_PATH=$AB_HOME/lib/postgresql/lib:$LD_LIBRARY_PATH
export PGUSER=opdb
export PGPASSWORD=abinitio
export PGPORT=6464
export PGDATABASE=OPDB
```

NOTE: These commands assume the default **operational database** name, username, and password that are used in **Control Center** and Operational Console standard installations. If your **operational database** uses different values, use those values instead of these defaults.

5. Enter the following command to restore the **operational database** from the backup file:

```
pg_restore --dbname=OPDB --clean --format=tar --verbose /new_path/
dumpfile
```

where:

- *new_path* — The filesystem path on the new **Co>Operating System** host to which you copied the database backup file.
- *dumpfile* — The name of the database backup file from which you want to restore the **operational database**.

6. Restart the **Control Center** application; for example:

```
ab-app start controlcenter
```

Upgrading an ab-db operational database on Co>Operating System 3.2.3 or later

NOTE: This section applies only to Control Center installations that use ab-db as the operational database server. If you do not run your operational database on ab-db, skip the remainder of this section.

Co>Operating System Versions 3.2.3 and later include a new version of ab-db database server software that is incompatible with the ab-db software that was included with older versions of the Co>Operating System. Depending on your current Co>Operating System and operational database configuration, you may need to use the `abdb_server_upgrade.ksh` script, included in the Control Center 3.2.3 product bundle, to upgrade your current operational database to make it compatible with the newer version of ab-db software.

This section contains the following topics:

- Determining if an operational database upgrade is required
- Understanding the ab-db compatibility issue
- Understanding the `abdb_server_upgrade.ksh` script
- Co>Operating System requirements for running the `abdb_server_upgrade.ksh` script
- Using the `abdb_server_upgrade.ksh` script

Determining if an operational database upgrade is required

You must upgrade your operational database to make it compatible with the latest ab-db version if both of the following conditions are true:

- Your existing operational database was created using an ab-db version from Co>Operating System Version 3.2.2 or earlier.
- Your Co>Operating System was upgraded to Version 3.2.3 or later after the operational database was created with ab-db on an older Co>Operating System.

The Control Center `setup.sh` and the `upgrade_database.ksh` scripts automatically check for an incompatible ab-db operational database in your current installation. If an incompatible operational database is found, you are prompted to run the `abdb_server_upgrade.ksh` script before proceeding with the remainder of the Control Center upgrade.

If you do not run your operational database on ab-db, or if you perform the Control Center upgrade on Co>Operating System Version 3.2.2 or earlier, then the ab-db compatibility issue does not apply.

Understanding the ab-db compatibility issue

The ab-db database server software that is included with the Co>Operating System is based on PostgreSQL. In Co>Operating System Versions 3.2.2 and earlier, the base implementation is PostgreSQL version 8.x. In Co>Operating System Versions 3.2.3 and later, the base implementation is PostgreSQL 9.x. The differences between PostgreSQL server 8 and 9 are significant, and the databases that are created by the different server versions are incompatible with each other.

TIP: You can use the following command to determine the PostgreSQL version on which your **ab-db** operational database was created:

```
ab-db sql OPDB --version
```

Understanding the abdb_server_upgrade.ksh script

The **db**s directory in the **Control Center** 3.2.3 product bundle includes a script named **abdb_server_upgrade.ksh**, which converts an **ab-db** database based on PostgreSQL 8.x to a format that is compatible with PostgreSQL 9.x. Note the following:

- If an upgrade conversion is required for your existing **ab-db** operational database, the **Control Center setup.sh** and **upgrade_database.ksh** scripts automatically prompt you to run the **abdb_server_upgrade.ksh** script before proceeding with the remainder of the **Control Center** upgrade.
- If neither the **setup.sh** nor **upgrade_database.ksh** script prompts you to run the **abdb_server_upgrade.ksh** script, then the **ab-db** upgrade conversion is not required for your **Control Center** upgrade.
- The **setup.sh** and **upgrade_database.ksh** scripts only detect upgrade requirements for **ab-db** databases. The scripts do not check or test generic PostgreSQL databases that are not implemented as **ab-db** database instances.

The **abdb_server_upgrade.ksh** script performs the following tasks:

1. **pg_dump** — The script uses the **\$AB_HOME/lib/postgresql/bin/pg_dump** command in the Version 3.2.2 or older **Co>Operating System** to create a backup of the existing operational database. The database backup is created in the **\$AB_DATA_DIR/ops/dbbackup** directory.
2. **createdb** — The script uses the **\$AB_HOME/lib/postgresql/bin/createdb** command in the Version 3.2.3 or later **Co>Operating System** to create a new operational database instance.
3. **pg_restore** — The script uses the **\$AB_HOME/lib/postgresql/bin/pg_restore** command in the Version 3.2.3 or later **Co>Operating System** to restore the data from the backup that was created in the first step. When the **pg_restore** command writes the data to the new **ab-db** database instance, the data is written in a format that is compatible with PostgreSQL 9.x.

Co>Operating System requirements for running the abdb_server_upgrade.ksh script

In order to use the **abdb_server_upgrade.ksh** script, you must have access to the older (pre-3.2.2) **Co>Operating System** on which the existing **ab-db** operational database instance is running, and the newer (3.2.3 or later) **Co>Operating System** on which the **Control Center** upgrade will be performed.

WARNING! When you or someone else upgrades your **Co>Operating System** to Version 3.2.3 or later, be sure to retain the older **Co>Operating System** until after you have migrated all **ab-db** and **ab-app** instances. If you no longer have access to the older **Co>Operating System** on which your **ab-db** and **ab-app** instances were running, contact **Ab Initio Support** for assistance.

Using the `abdb_server_upgrade.ksh` script

Use the following procedure if either the [Control Center `setup.sh`](#) or [`upgrade\_database.ksh`](#) script prompts you to run the [`abdb\_server\_upgrade.ksh`](#) script. Do not run the [`abdb\_server\_upgrade.ksh`](#) script if you are not prompted to do so.

► **To run the `abdb_server_upgrade.ksh` script:**

1. Ensure that the [ab-db operational database](#) instance that you want to upgrade is running on the older (pre-3.2.2) [Co>Operating System](#).

The [ab-db](#) instance must be running order for the script's [pg_dump](#) command to work.

2. Ensure that the `AB_HOME` configuration variable points to the new (3.2.3 or later) [Co>Operating System](#) on which you are performing the [Control Center](#) upgrade.
3. In a command shell, change to the [controlcenter-3.2.3/dbs](#) directory.
4. Launch the [abdb_server_upgrade.ksh](#) script:

```
./abdb_server_upgrade.ksh
```
5. When prompted, enter the required values:

- Database instance name (by default, `OPDB`)
- Database password (by default, `abinitio`)
- `AB_HOME` directory for the older (pre-3.2.2) [Co>Operating System](#) on which the existing [ab-db](#) instance is running
- `AB_HOME` directory for the new (3.2.3 or later) [Co>Operating System](#) on which you are upgrading the [Control Center](#)

The script performs the database upgrade, and writes status output to the terminal.

6. When the script finishes, proceed normally with the remainder of the [Control Center](#) upgrade procedures described in this chapter.

Upgrading the [operational database](#) from versions prior to 3.2.2

NOTE: This section applies only to upgrades of the [operational database](#) from Version 3.2.1 and earlier. Upgrades from [operational database](#) Version 3.2.2 and later do not have the limitations that are described in this section.

Version 3.2.2 of the [Control Center](#) features significant [operational database](#) changes that provide two major benefits:

- **Greatly improved History view performance** — Unlike previous releases, version 3.2.2 of the [Control Center](#) precalculates time data that facilitates aggregation of job metric values. The calculations are made once in bulk during upgrade, and incrementally thereafter. The time data is incorporated in the [operational database](#), reducing from minutes to seconds the time required to generate History view charts for capacity planning.
- **Prevention of a potential scalability problem** — In release 3.2.1 of the [Control Center](#), primary keys were stored as 32-bit integers in the [operational database](#). After a few years, at facilities that process hundreds of thousands of jobs each day and where the [Control Center](#) tracks

perhaps a dozen metrics for each job, running out of keys would become a practical concern. To prevent this, the upgrade to version 3.2.2 of the [Control Center](#) recasts all primary keys as 64-bit integers.

These [operational database](#) enhancements require the database upgrade script to do the following:

- Convert all 32-bit keys to 64 bits
- Add new database columns to accommodate new job metric data
- Precalculate time data for metric value aggregation

What you should expect

The need to perform this work makes the upgrade from [operational database](#) Version 3.2.1 and earlier to Version 3.2.2 or later a lengthy one. Exactly how lengthy depends on several factors — principally, the following:

- The number of jobs in your Version 3.2.1 or earlier database
- The database management system (DBMS)
- Available processing resources

Based on internal testing, we expect a typical upgrade involving a database of 100,000 jobs to take about 30 minutes. The time scales linearly with the number of job records that must be processed, so if your database holds a million job records, you should expect a wait of several hours.

Recommended limits

If your [operational database](#) falls within the following limits, proceed with the upgrade. If your database exceeds the limits, see "[Alternatives for upgrading larger 3.2.1 and earlier databases to 3.2.3](#)" before continuing:

- **General** — As a practical matter, *if your database holds much more than a million jobs*, you should purge older and less essential job and job metric data before you start the upgrade.
- **DB2 databases** — Testing shows that DB2 databases perform significantly worse than other supported databases. Because of this, *do not upgrade a DB2 database that contains more than 100,000 jobs*. Either purge the DB2 database of excess jobs, or start over with a fresh installation of [Control Center](#) Version 3.2.3.

For specific questions or concerns about the upgrade process or limits, contact [Ab Initio Support](#).

Alternatives for upgrading larger 3.2.1 and earlier databases to 3.2.3

If the [operational database](#) that you want to upgrade to 3.2.2 includes more jobs than the upgrade process can practically accommodate, consider the alternatives summarized in the following table and described below:

Alternative	Benefits	Drawbacks
Back up and purge the database before upgrading	Quick	No job history is retained
Back up the database and purge older records before upgrading	Some job history is retained	- Takes longer - Must decide what records to purge

Back up and purge the database before upgrading

► To back up and purge the database:

1. Perform a full backup of your 3.2.1 or earlier [operational database](#).
For instructions, see the documentation accompanying your DBMS software.
2. Purge all job records.
3. Perform the upgrade to version 3.2.3, as described later in this chapter.
4. Restore job records from backup as required.
For instructions, see the documentation accompanying your DBMS software.

Back up the database and purge older records before upgrading

► To back up the database and purge older records:

1. Perform a full backup of your 3.2.1 or earlier [operational database](#).
For instructions, see the documentation accompanying your DBMS software.
2. Purge older or less essential job records so that what remains falls within [recommended limits](#).
3. Perform the upgrade to version 3.2.3, as described later in this chapter.
4. (Optional) Restore additional job records from backup as required.
For instructions, see the documentation accompanying your DBMS software.

Adjusting abinitiorc and apphubrc settings

When you install [Ab Initio](#) Version 3.2.x server software on a computer that already has an [Ab Initio](#) Version 3.1.x Application Hub link, the [Ab Initio](#) 3.2.x installer copies the contents of the 3.1.x [Co>Operating System](#) \$AB_HOME/[config/abinitiorc](#) configuration file into the new Application Hub configuration file, \$AB_APPLICATION_HUB/[config/apphubrc](#). However, the 3.1.x [abinitiorc](#) settings are not copied into the [Co>Operating System](#) Version 3.2.x [abinitiorc](#) file.

NOTE: [Ab Initio](#) server software upgrades do not modify the contents of any per-user \$HOME/[.abinitiorc](#) files.

Because most of the [Control Center](#) and Operational Console configuration settings for a monitored [Co>Operating System](#) host are kept in the \$AB_HOME/[config/abinitiorc](#) file for the host rather than in the \$AB_APPLICATION_HUB/[config/apphubrc](#) file, you must move those configuration settings from the [apphubrc](#) file back into the [abinitiorc](#) file after you upgrade from [Ab Initio](#) Version 3.1.x software to Version 3.2.x.

Specifically, you must move any entries for the following configuration variables from the \$AB_APPLICATION_HUB/**config/apphubrc** file back into the \$AB_HOME/**config/abinitiorc** file:

- AB_DATA_DIR
- AB_OPS_ALLOW_TENTATIVE_RESOURCE_REQUESTS
- AB_OPS_CONSOLE_URL
- AB_OPS_FILE_PATTERN_MATCH_LIMIT
- AB_OPS_FILE_STABILIZATION_INTERVAL
- AB_OPS_FSYNC_EVERY_EVENT
- AB_OPS_LOCAL_HOSTNAME
- AB_OPS_LOG_TENTATIVE_RESOURCE_REQUEST_WARNINGS
- AB_OPS_MONITOR
- AB_OPS_MONITOR_FILE_EVENT_LIMIT
- AB_OPS_MONITOR_GRAPH_DATASETS
- AB_OPS_MONITOR_RESOURCES
- AB_OPS_MONITOR_STDOUT_STDERR
- AB_OPS_MONITOR_SUBDIRECTORY
- AB_OPS_MULTUSER
- AB_USERNAME
- AB_WINDOWS_DOMAIN
- AB_WORK_DIR

The [Ab Initio](#) Version 3.2.x server software includes a program, \$AB_HOME/**bin/ab-filter-apphub-entries**, that can help you split the new 3.2 **apphubrc** contents into entries that belong in the **apphubrc** file and entries that belong in the **abinitiorc** file. After running this program, however, the [Co>Operating System](#) administrator should still examine the resulting **apphubrc** and **abinitiorc** files to make sure that each file has the right settings, and then make adjustments as necessary. For more information about these configuration files, see the following topics in the [Co>Operating System Administrator's Guide and Reference](#):

- [The system abinitiorc file](#)
- [The apphubrc file](#)
- [Configuring the Application Hub](#)