Sun Java™ System Migration Guide for Calendar Server for Microsoft Windows

Version 6 2005Q1

Part Number 819-4079-05

If you had an earlier version of Calendar Server (5.11 and earlier), after you install Calendar Server 6 2005Q1, and perform the post-installation configuration, you may need to migrate the component databases and the LDAP database.

A Choosing the Right Utilities section is provided in this chapter to assist you in choosing the correct utilities to run.

This guide contains the following sections:

Post-Installation Database Migration Utilities

After you have installed Calendar Server 6 2005Q1, if you have calendar databases and LDAP database entries from your earlier Calendar Server 5.1.1 installation, run the following utilities in the order given:

- `cs5migrate` or `cs5migrate_recurring`–Migrates your calendar databases from 5.x to 6.x format. These utilities are available for download from technical support.

If you plan to use the Connector for Microsoft Outlook and have recurring components, then use `cs5migrate_recurring`, which creates a master record and exceptions for each recurring series.

If you do not have recurring components in your existing database, or you do have them, but do not plan to use the Connector for Microsoft Outlook, use `cs5migrate`.

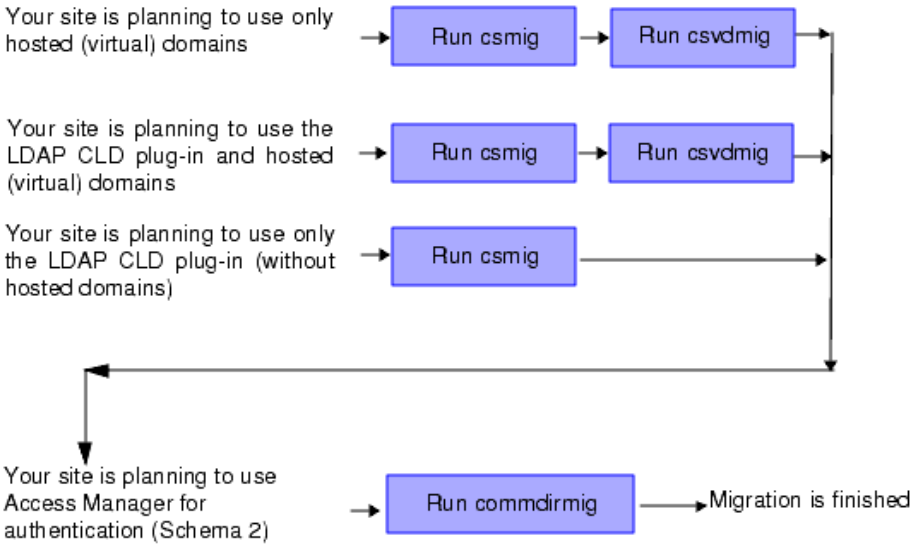
Both `cs5migrate` and `cs5migrate_recurring` are available only from technical support. They are not packaged with the product.

- csmig–Assigns an owner to each calendar in the Calendar Server 6.x database and maps each calendar ID (calid) to an owner, if needed, which allows support for hosted (virtual) domains and the LDAP Calendar Lookup Database (CLD) plug-in. This utility is packaged with Calendar Server. Run this utility after `cs5migrate` and before `csvdmig`.
- csvdmig–Upgrades a Calendar Server 6.x site to use hosted (virtual) domains by adding the calendar's domain (*@domainname*) to each calid. For example, in the domain `sesta.com`, jdoe's calid would now be *jdoe@sesta.com*. This utility is packaged with Calendar Server. Run this utility after `cs5migrate`, and `csmig`.
- commdirmig–Migrates LDAP data from Schema 1 to Schema 2 in preparation for use with Access Manager 6.1 or higher. This utility is packaged with Access Manager.

Choosing the Right Utilities

Since there are so many potential utility choices, Figure 1 shows different configuration scenarios and which utilities to run and in what order.

Figure 1 Choosing Migration Utilities to Run



csmig

The `csmig` utility assigns an owner to each calendar in the calendar database and maps each calendar ID (calid) to an owner, if needed.

The `csmig` utility supports hosted (virtual) domains and the LDAP Calendar Lookup Database (CLD) plug-in. Calendars in the migrated database are accessible using the LDAP CLD plug-in.

This section describes the following topics:

- csmig Functions
- csmig Requirements

- csmig Syntax
- csmig Migration Steps
- csmig Tips and Troubleshooting

NOTE If you are also moving from a non-hosted domain environment to a hosted domain environment, run `csmig` before `csvdmig`.

csmig Functions

The `csmig` migration utility performs these functions:

- `csmig` migrates both user and resource calendars in the current calendar database (`*.db files`) specified by the `caldb.berkeleydb.homedir.path` parameter. In the new destination target database, csmig updates entries required by the LDAP CLD plug-in in the calendar properties (`calprops`), events, todos (tasks), and group scheduling engine (GSE) database files.

  `csmig` writes only to the destination target database; it does does not update your existing calendar database.

- `csmig` updates LDAP attributes for all relevant LDAP entries, including icsSubscribed, `icsCalendar`, `icsCalendarOwned`, `icsFreeBusy`, `icsSet`, and for resource calendars, uid. `csmig` creates the `icsDWPHost` attribute for each calendar in the LDAP directory server database. `icsDWPHost` specifies the host name of the back-end server where a calendar resides.

- `csmig` assigns an owner to each calendar in the calendar database and maps each calendar ID (calid) to an owner, if needed. All default calids are kept as is, and no changes are made. Other calendars are mapped as follows:

  - User calendars that do not have valid owners will be owned by the user passed to csmig by the `-c option`. For example, if calendar ID jsmith does not have an owner, it will be converted to `orphan:jsmith`, where orphan is specified as the `-c option`.

  - Resource calendars that don't have an owner will be owned by the resource user passed to `csmig` by the `-r option`.

  - If a resource calendar has any colons (:) in the name, the colons are converted to underscores, so that the migrated name has only one colon.

For example, a calendar named football with owner bkamdar will be converted to bkamdar:football. A calendar named tchang:soccer with the owner bkamdar will be converted to bkamdar:tchang_soccer. A resource calendar named auditorium:room1 with an owner admin1 will be converted to admin1:auditorium_room1.

csmig Requirements

The requirements for using csmig are:

- The calendar database must not be corrupted. Use the csdb check command to check your calendar database, and if necessary, run the csdb rebuild command to rebuild the database.
- You must have sufficient disk space for the new destination target database and if applicable, your backup database.
- To run csmig, log in as icsuser (or as the Calendar Server runtime user ID specified during configuration). If you run csmig as superuser (root), you might need to reset the permissions for the migrated files.

  You must also have privileges to manage the attributes of calendar users in the LDAP directory server that stores user preferences.
- Calendar Server must be stopped.

csmig Syntax

The csmig utility has the following syntax:

```
csmig [ -t DestinationDB ] [ -b Backend-DWPHost ]
      [ -o OutputFile ]  [ -e ErrorFile ] [ -m MappingFile ]
        -c calendarOwner -r resourceOwner { migrate|dryrun }
```

The following table lists the utility options, gives a description of each, and gives the default value.

Table 1 Options for csmig

csmig Options	Description and Default Value
-t DestinationDB	Specifies the destination target database that csmig generates. The default is MigratedDB.
-b Backend-DWPHost	Specifies the name of the DWP back-end host server. This name must match the DWP back-end host server name specified in the ics.conf file.

Table 1 Options for `csmig` *(Continued)*

csmig Options	Description and Default Value
`-o OutputFile`	Specifies an output file that captures the `csmig` output to the screen as well as any errors that occur. The default is `MigrateOut`.
`-e ErrorFile`	The file where csmig writes any errors or database entries that cannot be resolved. If database entries cannot be resolved, they are not written to the destination database. The default is `MigrateError`.
`-m MappingFile`	Specifies an output `mapping file` generated in `dryrun` mode that lists entries in the LDAP schema that need to be changed. For example: Old calid = `jsmith`; New calid = `jsmith:basketball` The `mapping file` provides only a list of changes to make to the LDAP schema. `csmig` does not actually make the changes to the schema. The `mapping file` is not used in migrate mode.
`-c calendarOwner`	Specifies the owner for user calendars that don't have owners.
`-r resourceOwner`	Specifies the owner for resource calendars that don't have owners.
`migrate|dryrun`	Specifies which mode the utility is running in. Use migrate mode to perform the migration. Use `dryrun` mode to generate the output `mapping file` before you actually migrate.

csmig Migration Steps

After you have installed and configured Calendar Server 6.x, you must run `csmig` to migrate your existing Calendar Server and LDAP data. Migration of the LDAP data is required for the LDAP CLD plug-in to work properly. Use these steps to migrate calendar data using `csmig`:

1. Configure your Directory Server using `comm_dssetup.pl`.

   If you have not already indexed LDAP attributes using `comm_dssetup.pl`, do so at this time. This will greatly help performance of the LDAP data migration.

2. Using a staging server (not your production server), perform a test dry run.

   A dry run reports what `csmig` would do during an actual migration but does not migrate any data. After the dry run, and before you actually migrate, correct any errors and determine a plan to handle any unresolved calendars.

   For instructions on how to perform a test dry run, see To Perform a Test Dry Run.

3. Migrate Your Production Data

   During a production run, `csmig` migrates the calendar database (`.db files`) and LDAP data (user and group preferences data), `icsSubscribed`, `icsCalendar`, `icsCalendarOwned`, `icsFreeBusy`, `icsSet`, and `uid` (for resource calendars). After the migration, all calendar resources will have an LDAP entry created.

   For instructions on how to migrate your production data, see To Migrate Your Production Data.

To Perform a Test Dry Run

1. Install Calendar Server 6.x (if necessary) on the staging server.
2. Copy a snapshot of your calendar database to the staging server.
3. Mimic your production LDAP environment on the staging server by performing the following tasks:
   - Install Directory Server.
   - Install a snapshot of the LDAP database on this server.
4. Run `comm_dssetup.pl` to configure the staging Directory Server.
5. Run `csconfigurator.bat` to configure the staging Calendar Server.
6. Log in as `icsuser` (or, if its different, log in as the Calendar Server runtime user ID specified during configuration).
7. Change to the *jes_installdir*`/CalendarServer/bin` directory.
8. Run the `csdb check` command to check your database for corruption. If corruption is indicated, run `csdb rebuild` to rebuild the database.
9. Consider creating a catchall calid for user calendars that don't have an owner. For example, the following command creates a user with the calid of orphan:

   `csuser -g orphan -s adminuser -y` *password* `-l en -c orphan create orphan`

10. Stop the Calendar Server using the `stop-cal` command (if necessary).

    jes_installdir`/CalendarServer/bin/stop-cal.bat`

11. Run `csmig` with the dryrun option. For example, you might enter:

    ```
    csmig -b sesta.com -o csmig.out -e csmig.errors -m csmig.map -c
    orphan -r calmaster dryrun
    ```

    This command assigns user calendars without an owner (orphan calendars) to the owner orphan and resource calendars without an owner to the owner calmaster.

12. Check the output mapping file (`csmig.map`). The mapping file lists entries that need to be updated in the LDAP schema.

13. Check the output, mapping, and error files. Resolve any LDAP issues or errors that you find. Determine how you will handle any unresolved calendars before the actual migration. Several options are:

    - Delete any unneeded calendars before you migrate.
    - Assign owners to any unresolved calendars.
    - Allow `csmig` to assign owners to the calendars during migration using the -c and -r options.

14. Run `csmig` to migrate your staging calendar database.

    For example, the following command migrates the calendar database to the *jes_installdir*`/CalendarServer/testcsdb/` directory:

    ```
    csmig -t jes_installdir/CalendarServer/testcsdb/ -b sesta.com -o
    csmig.out -e csmig.errors -m csmig.map -c orphan -r calmaster
    migrate
    ```

15. After the test migration is finished, perform these steps to check out the newly migrated calendar database.

    a. Copy the migrated database to the `/csdb` directory specified by the `caldb.berkeleydb.homedir.path` parameter. Or, edit this parameter to point to the new location of the migrated database.

    b. Run `csdb check` on the new calendar database. The number of events and todos in the migrated database should match the pre-migration totals.

    c. Search for `icsCalendarOwned` entries and make sure that the entries match the pre-migration number of calendars.

    d. Log in to Calendar Express or Communications Express and verify some of the calendars in the migrated database.

If the test migration is successful, you are ready to migrate your production database.

To Migrate Your Production Data

1. Log in as `icsuser` (or as the Calendar Server runtime user ID specified during configuration).
2. Change to the *jes_installdir*`/CalendarServer/bin` directory.
3. Stop the Calendar Server using the stop-cal command (if necessary).

   jes_installdir`/CalendarServer/bin/stop-cal.bat`
4. Backup the following data:
   - Calendar database (`.db files`).
   - LDAP `data: slapd` database directory and LDAP database.
   - `ics.conf` file. This step is not actually required, but it can be useful if you need to revert to your original configuration.
5. Run `csmig` with the migrate option.

   For example, the following command migrates the calendar database to the *jes_installdir*`/CalendarServer/newcsdb/` directory:

   ```
   csmig -t jes_installdir/CalendarServer/newcsdb/ -b sesta.com -o csmig.out
   -e csmig.errors -m csmig.log -c orphan -r calmaster migrate
   ```
6. Check for any unresolved calendars in the error file (`csmig.errors`) and resolve them according to your plan from Step 13 under To Perform a Test Dry Run.
7. Run the `csdb check` command to check your migrated database. If any corruption is indicated, run `csdb rebuild` to rebuild the database.
8. Copy the new migrated database to the `/csdb` directory specified by the `caldb.berkeleydb.homedir.path` parameter. Or, edit this parameter to point to the new location of the migrated database.
9. Enable the LDAP CLD plug-in by making any necessary changes to the following configuration parameters in the `ics.conf` file:
   - `service.dwp.enable = "yes"`
   - `service.dwp.port = "9779"`
   - `csapi.plugin.calendarlookup = "y"`
   - `csapi.plugin.calendarlookup.name = "*"`
   - `caldb.cld.type = "directory"`

- `caldb.dwp.server.default = "default-server-name"`
- `caldb.dwp.server.server-hostname.ip = "server-hostname"` (for each back-end server including the local server)
- `caldb.cld.cache.enable = "yes"` (if you want to use the CLD cache option)
- `caldb.cld.cache.homedir.path` specifies the location of the CLD cache directory. The default is *jes_installdir*`/CalendarServer/csdb/cld_cache`.

10. Restart Calendar Server using the `start-cal` command.
11. Log in to your calendar user interface (Calendar Express or Communications Express) and verify that your configuration is working by checking several of the migrated calendars.

    To disable alarms while you are making your checks, set each of the following parameters in the `ics.conf` file to "no":

    - `caldb.serveralarms = "no"`
    - `caldb.serveralarms.dispatch = "no"`
    - `service.ens.enable = "no"`
    - `service.notify.enable = "no"`
    - `ine.cancellation.enable = "no"`
    - `ine.invitation.enable = "no"`
    - `service.admin.alarm = "no"`

csmig Tips and Troubleshooting

The section describes the following tips and trouble shooting examples:

- The csmig dry run calendar shows the wrong owner for a calendar.
- The LDAP calendar search doesn't work correctly.
- The csmig dry run indicates duplicate calendar names.
- How do I assign orphan calendars to different owners?
- How do I move calendar users to another back-end server?

The csmig dry run calendar shows the wrong owner for a calendar.

Example Problem

A calendar named `tchang:myCalendar` has the owner jsmith in the calendar database, and the `csmig` dry run shows the mapping as `jsmith:tchang_myCalendar`. However, you would like to name this calendar `tchang:myCalendar` and assign the owner as tchang.

Example Solution

Before the migration, use the cscal utility to change the owner of the calendar `tchang:myCalendar` to `tchang`. Once this is done, the migration will map this calendar to `tchang:myCalendar` and add `icsCalendarowned` to the LDAP entry for user ID `tchang`.

The LDAP calendar search doesn't work correctly.

Problem

After migration, the LDAP calendar search is enabled, but the calendar search dialog does not return any results or returns only partial results.

Solution

Enabling the LDAP calendar search allows Calendar Server to search `(&(objectclass=icscalendaruser)(icscalendarowned=*substr*))`.

Manually run two different searches on the LDAP data with the following filters and compare the output:

- ldapsearch with filter `(&(objectclass=icscalendaruser)(icscalendarowned=*substr*))`
- ldapsearch with filter `(icscalendarowned=*substr*)`

Since the server uses the filter that includes `icsCalendarUser` objectclass, the LDAP server might have been deployed with the schema check disabled, and some calendar entries may have been provisioned without the `icsCalendarUser` objectclass.

The csmig dry run indicates duplicate calendar names.

Example Problem

The `csmig` dry run mapping file and output file indicate that there is a duplicate calendar name. For example, in the original database, `jsmith` owns the following calendars:

- `basketball` with 5 events
- `jsmith:basketball` with 10 events

The dry run indicates that during a migration, the two calendars will be merged, and the resulting calendar will be `jsmith:basketball` with owner `jsmith` and 15 total events

The output file will include the following warning message:

Error modifying calendar properties, error=2

Example Solution

If you don't want the two calendars to be merged, change the owner of basketball to a user other than jsmith before the migration. This will preserve the data integrity of the two separate calendars.

How do I assign orphan calendars to different owners?

Problem

By default `csmig` assigns all orphan calendars to a single owner, but I would like to assign different owners for some orphan calendars.

Solution

`csmig` does not accept the `mapping file` in the command line. However, you can assign owners to the orphan calendars in the original database before the migration. Check the dry run mapping file for all orphan calendars. Then use the `cscal` utility to assign owners to the orphan calendars before the migration. Run `csmig` in dryrun mode again to verify the new owners.

How do I move calendar users to another back-end server?

Problem

How do I move users from one back-end server to another?

Solution

To move a calendar user, you export each of the user's calendars on the original server and then import the calendars on the second server. After the calendars are moved, you can delete the calendars on the original server.

csvdmig

The `csvdmig` utility modifies the Calendar Server database and LDAP directory server database for sites that want to use hosted (virtual) domains.

This sections contains the following topics:

csvdmig Functions

The `csvdmig` utility adds the domain name to the user ID as follows:

- The format of calendar IDs (calids) is changed:

  From: `userid[:calendar-name]`

  To: `userid@domain[:calendar-name]`

- Access Control List (ACL) access rules are changed:

  From: `userid`

  To: `userid@domain`

- The LDAP directory server user entries for the Calendar Server attributes are modified as:

  `userid[:calendar-name]` to `userid@domain[:calendar-name]`.

- Updates the owner and attendee fields in events and tasks in the calendar database.

  For example: if `jsmith` in the domain `sesta.com` is the owner of an event, the new owner field would contain `jsmith@sesta.com`.

CAUTION The `csvdmig` utility updates the databases and LDAP directory in place. That is, it does not create a separate migrated database, but alters the database you are converting. Therefore, to be safe, run `csvdmig` against snapshots of your databases and LDAP directory.

csvdmig Syntax

The `csvdmig` utility has the following syntax:

```
csvdmig [-t DestinationDB] [-c ConfigFile] [-e ErrorFile] [-m
MappingFile]

migrate [DB | LDAP]
```

The following table lists the options used by `csvdmig`, and gives a description of each.

Table 2 Options for `csvdmig`

Option	Description and Default Value
`-m MappingFile`	Input parameter specifying a mapping file. For more information on the mapping file, see Mapping File. The default is `MigrateMapping`.
`-c ConfigFile`	Input parameter that specifies a Calendar Server configuration file. The default is the `ics.conf` file.
`-t DestinationDB`	Output parameter that specifies the location of the database. The default is `MigratedDB`. See Destination DB.
`-e ErrorFile`	Output parameter that specifies the name of the error file for errors that cannot be resolved. The default is `MigrateError`.
`DB | LDAP`	Specifies which database to modify: DB–the Calendar Server database LDAP–the LDAP directory The default is the calendar database (`DB`).

Mapping File

The `mapping file` is an input text file that maps existing users to their respective domains. You must create the mapping file before you run `csvdmig`. Specify one entry per line with a space between the old and new values. For example:

```
user1 user1@sesta.com

user2 user2@siroe.com

user3 user3@sesta.com

...

user-n user-n@siroe.com
```

Destination DB

Even though the variable is named *DestinationDB* and the default is MigratedDB, `csvdmig` does not create a separate migrated database. It updates in place the original database that you specify with this option.

csvdmig Examples

- Migrate the LDAP directory server data using default values:

  ```
  csvdmig migrate LDAP
  ```

- Migrate the Calendar Server database:

  ```
  csvdmig -t targetDB -e errorFile -m mappingFile migrate
  ```

commdirmig

The `commdirmig` utility migrates your LDAP data from Sun LDAP Schema 1 to Schema 2 in preparation for using Access Manager for authentication services.

This section contains the following topics:

Who Should Run the Utility

If you previously used Messaging Server 5.x or Calendar Server 5.x, your LDAP entries were formatted Schema 1. In your new Calendar Server 6 2005Q1 environment, if you are going to use Access Manager for authentication, you must convert your LDAP entries to Schema 2 format by running this utility.

If you are not using Access Manager, you should still consider migrating your LDAP data, since Schema 2 is the preferred LDAP mode for all Java Enterprise System products that use LDAP. In the future it is possible that newer versions of the communications products (Calendar, Messaging and Instant Messaging) may not support Schema 1. However, the migration can be deferred until a later, more convenient time if you are not going to be using Access Manager at this time.

NOTE If you have a separate LDAP directory for preferences, you must run `commdirmig` on that LDAP as well as the one used for authentication.

When to Run the Utility

If you are migrating from a pre-Java Enterprise System version of Calendar Server, run this utility after you run `cs5migrate`, `csmig` and `csvdmig`.

Where to Find Documentation

This migration utility requires special preparation and planning. It is documented in *Sun Java System Communications Services Schema Migration Guide.*

Where to Find the Utility

For Sun Java Enterprise System 2005Q1, this utility is bundled with Access Manager 2005Q1, along with the user management utility, `commadmin`.

If you are not updating Access Manager and just need the migration utility for your Calendar Server, a patch is available from technical support for just the utility.