

Backup

1. Recommended Backup Strategy

One approach to backup is to take regular exports of the database table, or copies of the Javaspaces files, which contain the entire Playwright world. This is certainly the simplest approach to implement.

However, such an approach does not allow for *incremental backup* - saving only those changes made since the last backup. Hence it is costly in space on backup media, and in the time taken to write each backup.

An alternative approach which allows for incremental backup of a RADRunner system is to export the entire world to Playwright. This can be done from the 42 Role instance via the *Boss* user, for example, or from another role instance with similar functionality. The export will result in a directory containing each object in the world as an XML text file, and this directory can then be backed up using any operating system level backup tool.

If your world is very large, the export may take several minutes. It is best in this case to ensure that the session timeout on your HTTP server is longer than the export requires. If the timeout is not long enough, *the export will still succeed*, but you will lose your connection to RADRunner and not receive any confirmation of this. A practical setting for HTTP timeout with a large world would be 20 minutes, for example. See [Automating Backup](#) for an alternative approach which removes this issue.

To back up the directory, it is preferable to use an operating system level backup tool which supports incremental backup, so that only those objects in the world which have been inserted or updated since the previous backup will be written to secondary storage.

The directory should be cleared down after each run of the operating system backup tool, so that the operating system level backup tool can detect when objects have been deleted from the world.

2. Restore

With the simple, non-incremental approach to backup, restore can be effected by importing the desired version of the database table, or replacing the Javaspaces files with a specific copy.

With the incremental approach to backup, restore can be done by using the operating system level backup tool to restore a particular version of the backup directory, and:

- *either* importing all the files in a particular version of the directory to restore the entire world
- *or* importing specific Playwright files to restore individual objects as required.

The comment concerning [session timeout on your HTTP server](#) applies particularly to import. If you are doing an entire world import and the HTTP session times out before it is complete, you will not be able to save the import via RADRunner, and the imported objects will not be restored to the world. In general, import takes longer than export, so you may wish to increase the HTTP session timeout just before doing an import, then reset it to a more normal length afterwards.

Alternatively, if you do not wish to increase HTTP session timeout in this way, create an automated Role instance which contains an automated Activity to import the world from Playwright then set a flag. The Role instance should terminate on detecting that the flag is set. Assuming RADRunner automation is on for at least one application server, this Role instance will then execute in the background to carry out the import, without being constrained by an HTTP session.

3. Automating Backup

Since RADRunner provides sophisticated automation features, it is simple to schedule the world export to take place at a specific time every night, or once a week, for example. The operating system level backup tool (and the following directory clear down) could be scheduled via operating system facilities to take place an appropriate length of time afterwards - how long afterwards depends on the size of the world, and how long this takes to export in your RADRunner configuration.

This combination of techniques will result in fully automated backup of a RADRunner system. Moreover, there is then no issue concerning HTTP session timeout.