

# Configuration

## 1. Overview

RADRunner is largely a self-managing system - most configuration is done from within the application itself.

However, there are some areas which must be administered separately, for example:

- [Background automation of Role instances](#)
- [Logging categories and levels](#)
- [Clustering and load balancing](#)
- and so on.

Details of such administration activities are given in this section of the guide.

## 2. Administration Page

RADRunner does provide a basic administration web page, which can be used to:

- Start and stop the application within the application server
- Maintain the RADRunner [pseudo-session](#) mechanism

### 2.1. RADRunner Pseudo-sessions

The RADRunner pseudo-session mechanism is provided to remedy the default HTTP session tracking mechanism, which has the following problems:

1. Session objects are stored in memory and consume significant resources.
2. Session tracking relies on cookies. Some users turn off cookies, particularly for security reasons.
3. Session tracking uses session identifiers that are created by the server. In situations where many Web servers and many JVMs are used, the session tracking simply does not work because servers do not always understand each other's session identifiers.
4. In some versions of J2EE, still supported by RADRunner, it is not possible to detect when a session expires and send a remove() message to the EJBs that it uses. Thus, if users do not log off properly, the container ends up full of unnecessary EJBs which are only removed on expiry of a timeout.

Hence RADRunner implements the following alternative and optional mechanism:

1. Objects or values are not stored in memory but stored as text files. Each text file is associated with a particular user and the name of the text file serves as the session identifier. Therefore, the name is guaranteed unique.
2. The text files are stored in a special directory accessible by all Web servers. Hence the pseudo session can be used in a web farm without need for session affinity hardware or software.  
*Note that the session variables contain only references to EJBs (Handle instances), not the EJBs themselves, so the different web container JVMs must all use the same EJB container JVM for this to work.*
3. Session identifiers are not sent as cookies. They are encoded into URLs, which requires the rewriting of all hyperlinks, including the *action* attributes of HTML forms.
4. Each time the Front servlet finishes processing and dispatching a request, it checks for expired sessions and deletes them. When a session is deleted, it explicitly logs off all the API sessions it contains (which will remove the EJBs from the container) and destroys references to the API sessions (so that garbage collection can clean them up).

RADRunner pseudo-session support can be enabled by setting the system property `com.rolemodellers.rim.usePseudoSessions=true` (the default value is false).

This mechanism is robust, and incurs only a small performance penalty, but does entail a minor administration overhead - users who close their browsers without logging off RADRunner may leave obsolete session files on the server, and it is necessary to "clean the application" by clearing these down from time to time. Hence the administration web page provides a means of invoking this action. If pseudo-sessions are not enabled, invoking it will have no effect.

Note that the "clean the application" mechanism also has another purpose, which is to remove any cached XSL stylesheets from the application. By default, RADRunner caches in memory all stylesheets in compiled form, so as to reduce the overhead of XML/HTML generation during application usage. However, this means that if a stylesheet is updated after being used, the new version will not be picked up by users until RADRunner is restarted. In this case, cleaning the application will clear the cache and the latest stylesheet version will be picked up. It is also possible to turn off stylesheet caching altogether, by setting the system property `com.rolemodellers.rim.cacheStylesheets` to `false` - see [System Properties](#) for details of how to do this.