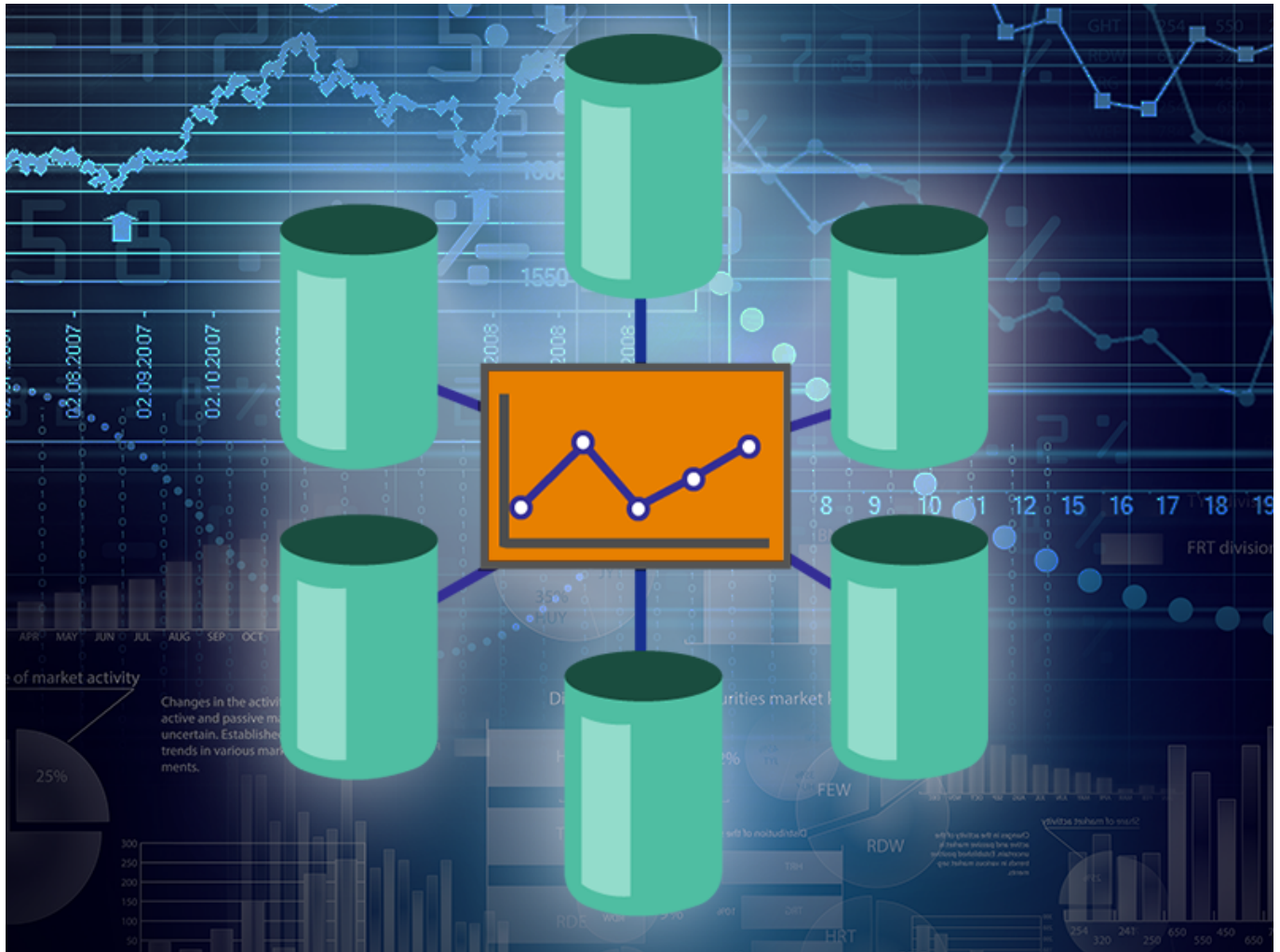


Article

[Alessandro Marin](#) · Apr 3, 2018 2m read

DeepSee: Databases, Namespaces, and Mappings - Part 2 of 5

The following post is a guide to implement a basic architecture for DeepSee. This implementation includes a database for the DeepSee cache and a database for the DeepSee implementation and settings.



Example 1: Basic architecture

Databases

This configuration for the Analytics server includes the APP-CACHE and APP-DEEPSEE databases. A crucial setting for DeepSee to run smoothly is that the DeepSee cache should never be journaled. Doing so will slow down the performance of the DeepSee engine, in addition to causing hyper-journaling and possibly problems with disk space. For this reason the DeepSee cache is stored in APP-CACHE, a separate DeepSee Cache database with journaling disabled.

APP-DEEPSEE is a DeepSee Implementation and Settings database containing the ^DeepSee.* globals. These globals define most of the DeepSee implementation such as cube definitions and settings, Cube Manager, user settings, and more. Note in the screenshot below that all databases are Read-Write and that it was decided to

enable Journaling only on the APP-DEEPSEE. It is recommended to journal this database since it contains all definitions, settings and user data.

Local Databases

Create New Database Refresh: ☐ off ☐ on 10 sec

Local Databases

The following is a list of the local databases:

Filter: Page size: 0 Max rows: 1000 Results: 12 Page: 1 of 1

Name	Directory	Size (MB)	Status	Resource	Encrypted	Journal	
CACHESYS	/home/amarin/intersystems/cache20172/mgr/	170	Mounted/RW	%DB_CACHESYS	No	Yes	- Globals
CACHELIB	/home/amarin/intersystems/cache20172/mgr/cachelib/	500	Mounted/RW	%DB_CACHELIB	No	No	- Globals
CACHETEMP	/home/amarin/intersystems/cache20172/mgr/cachetemp/	61	Mounted/RW	%DB_CACHETEMP	No	No	- Globals
CACHE	/home/amarin/intersystems/cache20172/mgr/cache/	11	Mounted/RW	%DB_CACHE	No	No	- Globals
CACHEAUDIT	/home/amarin/intersystems/cache20172/mgr/cacheaudit/	518	Mounted/RW	%DB_CACHEAUDIT	No	Yes	- Globals
APP-CACHE	/home/amarin/intersystems/cache20172/mgr/app-cache/	1	Mounted/RW	%DB_APP-CACHE	No	No	Delete Globals
APP-CODE	/home/amarin/intersystems/cache20172/mgr/app-code/	1	Mounted/RW	%DB_APP-CODE	No	No	Delete Globals
APP-DATA	/home/amarin/intersystems/cache20172/mgr/app-data/	1	Mounted/RW	%DB_APP-DATA	No	Yes	Delete Globals
APP-DEEPSEE	/home/amarin/intersystems/cache20172/mgr/app-deepsee/	1	Mounted/RW	%DB_APP-DEEPSEE	No	Yes	Delete Globals
DOCBOOK	/home/amarin/intersystems/cache20172/mgr/docbook/	225	Mounted/RW	%DB_DOCBOOK	No	No	Delete Globals
SAMPLES	/home/amarin/intersystems/cache20172/mgr/samples/	365	Mounted/RW	%DB_SAMPLES	No	No	Delete Globals
USER	/home/amarin/intersystems/cache20172/mgr/user/	11	Mounted/RW	%DB_USER	No	Yes	Delete Globals

Global Mappings

The following screenshot shows the mappings for this basic architectural implementation on the APP namespace. The ^DeepSee.Cache.* and ^DeepSee.JoinIndex map the DeepSee cache to the APP-CACHE database. The ^DeepSee.* globals maps, among others, DeepSee implementation and settings to the APP-DEEPSEE database.

Global Mappings

New Save Changes Discard Changes Cancel

Global Mappings

The global mappings for namespace APP are displayed below:

Filter: Page size: 0 Max rows: 1000 Results: 3 Page: 1 of 1

Global	Subscript	Database	
DeepSee.Cache*	APP-CACHE	Edit	Delete
DeepSee.JoinIndex*	APP-CACHE	Edit	Delete
DeepSee.*	APP-DEEPSEE	Edit	Delete

Comments

In this example for a basic architecture the DeepSee cache is stored in a dedicated database. This allows journaling to be disabled on the ^DeepSee.Cache* and ^DeepSee.JoinIndex globals.

Journaling of the APP-DEEPSEE database makes it possible to restore the DeepSee implementation (cubes, subject areas, DeepSee items, user settings, etc) in case of a disruptive event.

The configuration outlined in this example has some flaws. First, globals supporting synchronization are not taken care of. Second, the APP-DEEPSEE database also contains fact tables, indices, and other DeepSee globals. As a result, APP-DEEPSEE could become big in size making it not practical to journal and restore. This configuration can be considered acceptable if, for example, cubes do not contain a large amount of data.

In the [next example](#) of this series we will see how to map cube synchronization globals, fact tables and indices to separate databases.

[#Analytics](#) [#Beginner](#) [#Databases](#) [#Deployment](#) [#Mapping](#) [#Tutorial](#) [#InterSystems IRIS BI \(DeepSee\)](#)

Source URL: <https://community.intersystems.com/post/deepsee-databases-namespaces-and-mappings-part-2-5>