

Article

[Alessandro Marin](#) · Apr 17, 2018 2m read

## DeepSee: Databases, Namespaces, and Mappings - Part 4 of 5

The following post outlines a more flexible architectural design for DeepSee. As in the [previous example](#), this implementation includes separate databases for storing the DeepSee cache, DeepSee implementation and settings, and synchronization globals. This example introduces one new databases to store the [DeepSee indices](#). We will redefine the global mappings so that the DeepSee indices are not mapped together with the fact and dimension tables.



### Example 3: Fully flexible architecture

#### Databases

In addition to the databases defined before in the intermediate example, the Analytics server introduces the APP-INDEX database storing the indices, whereas the APP-FACT database only storing the fact and dimension tables. The reason to split indices from fact tables is that indices can possibly be big in size, and the block size for APP-FACT can be changed to improve performance.

As in the [previous example](#), enabling journaling for fact table and indices is optional. Please read the [note in the previous post](#) for a more extensive discussion.

Local Databases

Create New Database

Refresh: ☐ off ☐ on 10 sec

The following is a list of the local databases:

Filter: Page size: 0 Max rows: 1000 Results: 15 Page: 1 of 1

Name	Directory	Size (MB)	Status	Resource	Encrypted	Journal	
<a href="#">CACHESYS</a>	/home/amarin/intersystems/cache20172/mgr/	170	Mounted/RW	%DB_CACHESYS	No	Yes	- <a href="#">Globals</a>
<a href="#">CACHELIB</a>	/home/amarin/intersystems/cache20172/mgr/cache/lib/	500	Mounted/RW	%DB_CACHELIB	No	No	- <a href="#">Globals</a>
<a href="#">CACHETEMP</a>	/home/amarin/intersystems/cache20172/mgr/cache/temp/	11	Mounted/RW	%DB_CACHETEMP	No	No	- <a href="#">Globals</a>
<a href="#">CACHE</a>	/home/amarin/intersystems/cache20172/mgr/cache/	11	Mounted/RW	%DB_CACHE	No	No	- <a href="#">Globals</a>
<a href="#">CACHEAUDIT</a>	/home/amarin/intersystems/cache20172/mgr/cache/audit/	1	Mounted/RW	%DB_CACHEAUDIT	No	Yes	- <a href="#">Globals</a>
<a href="#">APP-CACHE</a>	/home/amarin/intersystems/cache20172/mgr/app-cache/	1	Mounted/RW	%DB_APP-CACHE	No	No	<a href="#">Delete</a> <a href="#">Globals</a>
<a href="#">APP-CODE</a>	/home/amarin/intersystems/cache20172/mgr/app-code/	1	Mounted/RW	%DB_APP-CODE	No	No	<a href="#">Delete</a> <a href="#">Globals</a>
<a href="#">APP-DATA</a>	/home/amarin/intersystems/cache20172/mgr/app-data/	1	Mounted/RW	%DB_APP-DATA	No	Yes	<a href="#">Delete</a> <a href="#">Globals</a>
<a href="#">APP-DEEPSEE</a>	/home/amarin/intersystems/cache20172/mgr/app-deepsee/	1	Mounted/RW	%DB_APP-DEEPSEE	No	Yes	<a href="#">Delete</a> <a href="#">Globals</a>
<a href="#">APP-DSTIME</a>	/home/amarin/intersystems/cache20172/mgr/app-dstime/	1	Mounted/RW	%DB_APP-DSTIME	No	Yes	<a href="#">Delete</a> <a href="#">Globals</a>
<a href="#">APP-FACT</a>	/home/amarin/intersystems/cache20172/mgr/app-fact/	1	Mounted/RW	%DB_APP-FACT	No	No	<a href="#">Delete</a> <a href="#">Globals</a>
<a href="#">APP-INDEX</a>	/home/amarin/intersystems/cache20172/mgr/app-index/	1	Mounted/RW	%DB_APP-INDEX	No	No	<a href="#">Delete</a> <a href="#">Globals</a>
<a href="#">DOCBOOK</a>	/home/amarin/intersystems/cache20172/mgr/docbook/	200	Mounted/RW	%DB_DOCBOOK	No	No	<a href="#">Delete</a> <a href="#">Globals</a>
<a href="#">SAMPLES</a>	/home/amarin/intersystems/cache20172/mgr/samples/	114	Mounted/RW	%DB_SAMPLES	No	No	<a href="#">Delete</a> <a href="#">Globals</a>
<a href="#">USER</a>	/home/amarin/intersystems/cache20172/mgr/user/	1	Mounted/RW	%DB_USER	No	Yes	<a href="#">Delete</a> <a href="#">Globals</a>

Global Mappings

The following screenshot shows the mappings for the implementation example above. The mapping for the ^DeepSee.Index global has been changed to be stored in the newly created APP-INDEX database. As in the intermediate example the mappings for the ^DeepSee.Fact\* and ^DeepSee.Dimension\* globals are still used to stored fact and dimension tables to the APP-FACT database. The query log and the last MDX query are optionally stored together with the DeepSee cache.

Global Mappings

New Save Changes Discard Changes Cancel

The global mappings for namespace APP are displayed below:

Filter: Page size: 0 Max rows: 1000 Results: 10 Page: 1 of 1

Global	Subscript	Database		
DeepSee.Cache.*		APP-CACHE	<a href="#">Edit</a>	<a href="#">Delete</a>
DeepSee.Dimension*		APP-FACT	<a href="#">Edit</a>	<a href="#">Delete</a>
DeepSee.Fact*		APP-FACT	<a href="#">Edit</a>	<a href="#">Delete</a>
DeepSee.Index		APP-INDEX	<a href="#">Edit</a>	<a href="#">Delete</a>
DeepSee.JoinIndex		APP-CACHE	<a href="#">Edit</a>	<a href="#">Delete</a>
DeepSee.LastQuery		APP-CACHE	<a href="#">Edit</a>	<a href="#">Delete</a>
DeepSee.QueryLog		APP-CACHE	<a href="#">Edit</a>	<a href="#">Delete</a>
DeepSee.Update		APP-DSTIME	<a href="#">Edit</a>	<a href="#">Delete</a>
DeepSee*		APP-DEEPSEE	<a href="#">Edit</a>	<a href="#">Delete</a>
OBJ.DSTIME		APP-DSTIME	<a href="#">Edit</a>	<a href="#">Delete</a>

## Comments

This example of architecture is the most flexible but requires creating five databases per namespace. As in the [second example](#) the DeepSee cache is mapped to a dedicated database with journaling disabled, and the synchronization globals are mapped to APP-DSTIME.

Mapping fact tables and indices allows DeepSee implementation and settings to be stored in a dedicated journaled database (i.e. APP-DEEPSEE), which can be easily used to restore the DeepSee implementation. Creating a separate database for indices is optional since in many cases it is sufficient to store the corresponding globals together with the fact tables in APP-FACT.

The last post of this series will contain a summary and list of databases used in our three examples.

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

---

Source URL: <https://community.intersystems.com/post/deepsee-databases-namespaces-and-mappings-part-4-5>