DeepSee: Databases, Namespaces, and Mappings - Part 3 of 5

Article
Alessandro Marin · Apr 10, 2018 3m read

DeepSee: Databases, Namespaces, and Mappings - Part 3 of 5

The following post outlines an architectural design of intermediate complexity for DeepSee. As in the previous example, this implementation includes separate databases for storing the DeepSee cache, DeepSee implementation and settings. This post introduces two new databases: the first to store the globals needed for synchronization, the second to store fact tables and indices.

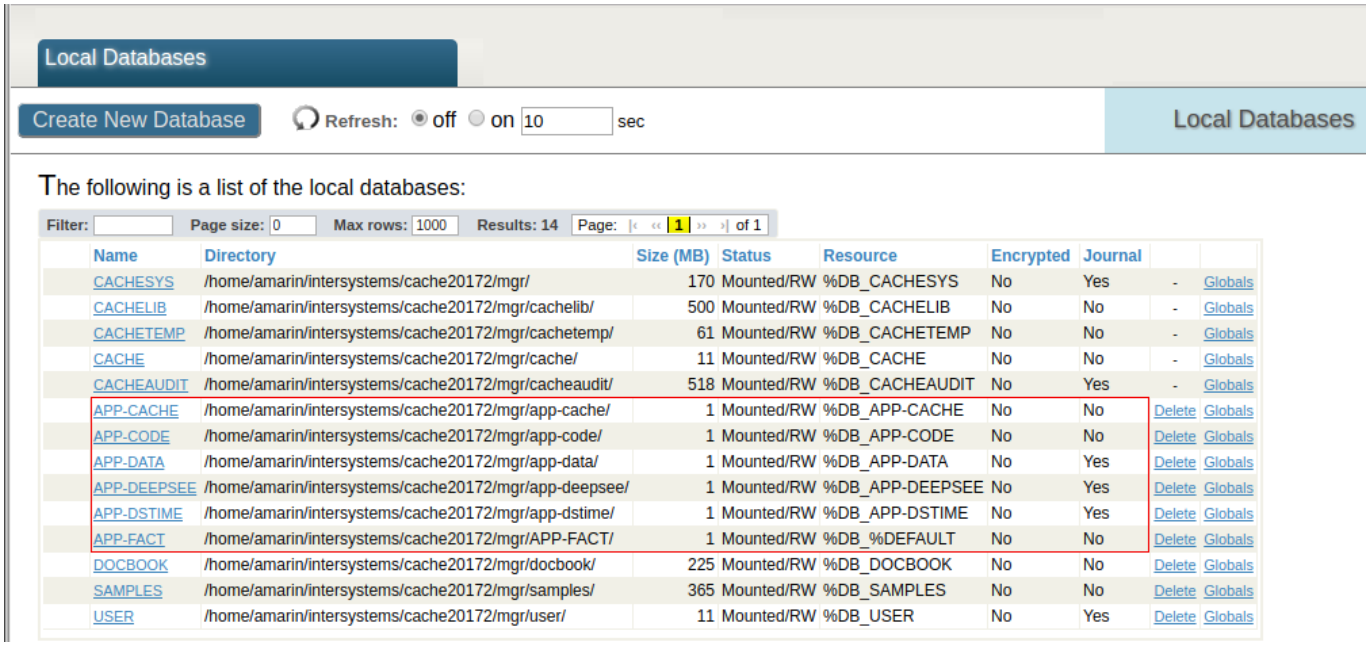
Example 2: More flexible design

Databases

In addition to the APP-CACHE and APP-DEEPSEE databases in the previous example, we define the APP-DSTIME and the APP-FACT databases.

The APP-DSTIME database contains the DeepSee synchronization globals ^OBJ.DSTIME and ^DeepSee.Update. These globals are mirrored from a (journaled) database on the Production server. Note that the APP-DSTIME database must be Read-Write in caché versions using ^DeepSee.Update.

The APP-FACT database stores the fact tables and indices. The reason to split indices from fact tables is that indices can possibly be big in size. By defining APP-FACT it is possible to have more flexibility with journal settings or to define a non-default block size. Enabling journaling for the APP-FACT database is optional. The choice mainly depends on whether Analytics can stay unavailable while rebuilding cubes in case of a disruptive event. In this example journaling fact tables and indices is disabled, and the typical reason for this choice is that cubes are small in size, build relatively fast, and undergo frequent periodical rebuilds. Please read the note on the bottom for a more extensive discussion.



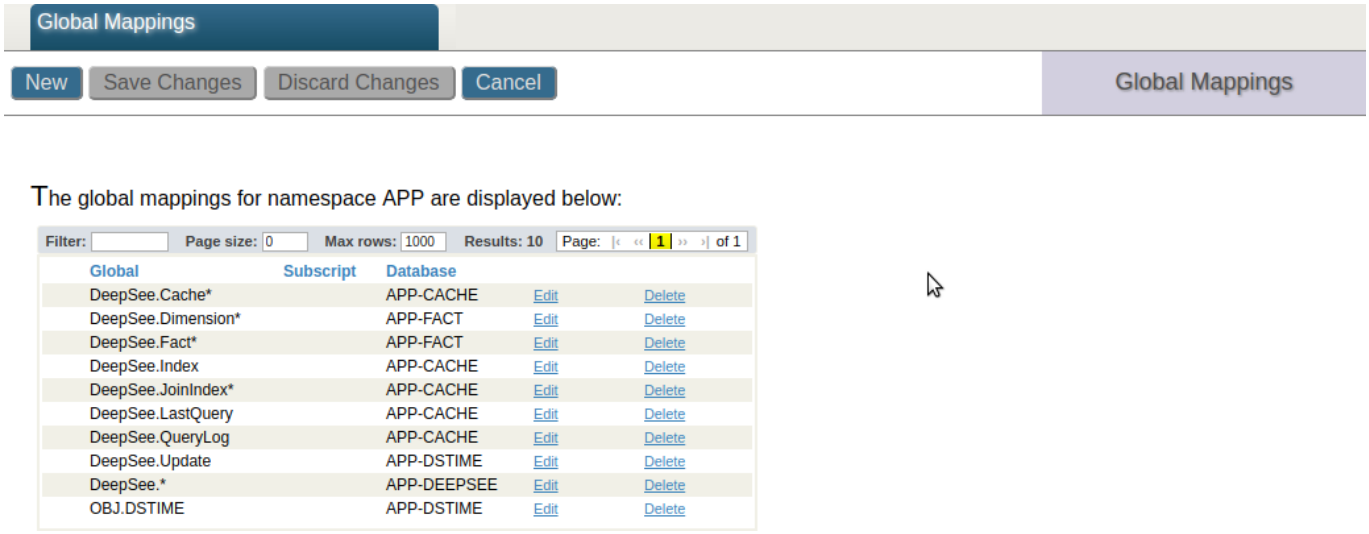
Name	Directory	Size (MB)	Status	Resource	Encrypted	Journal		
CACHESYS	/home/amarin/intersystems/cache20172/mgr/	170	Mounted/RW	%DB_CACHESYS	No	Yes	-	Globals
CACHELIB	/home/amarin/intersystems/cache20172/mgr/cachelib/	500	Mounted/RW	%DB_CACHELIB	No	No	-	Globals
CACHETEMP	/home/amarin/intersystems/cache20172/mgr/cachetemp/	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
APP-DSTIME	/home/amarin/intersystems/cache20172/mgr/app-dstime/	1	Mounted/RW	%DB_APP-DSTIME	No	Yes	Delete	Globals
APP-FACT	/home/amarin/intersystems/cache20172/mgr/APP-FACT/	1	Mounted/RW	%DB_%DEFAULT	No	No	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 the implementation example above.

The DeepSee synchronization globals ^OBJ.DSTIME and ^DeepSee.Update are mapped to the APP-DSTIME database. The globals ^DeepSee.LastQuery and ^DeepSee.QueryLog define a log for all executed MDX queries. In this example these globals are mapped to the APP-CACHE database together with the DeepSee cache. These mappings are optional.

The ^DeepSee.Fact* and ^DeepSee.Dimension* globals store the fact and dimension tables, whereas the ^DeepSee.Index global defines the DeepSee indices. These globals are mapped to the APP-FACT database.



Comments

As in the basic example the DeepSee cache is correctly stored in a dedicated database with journaling disabled. DeepSee implementation and settings are separately mapped to a journaled database to be able to restore the DeepSee implementation.

The globals supporting synchronization are mapped to APP-DSTIME and journaled on the Primary.

Mapping fact tables and indices to a dedicated database 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.

In the third and last example we will redefine the mappings for the APP-FACT database and create a database for the DeepSee indices.

Note on journaling and building cubes

Users should be aware that building cubes deletes and recreates the cubes' fact and index tables. This means that when journaling is enabled the SETs and KILLs of globals such as ^DeepSee.Fact*, ^DeepSee.Index are logged in journal files. As a result, rebuilding cubes might lead to a huge amount of entries in the journal files and possible problems with disk space.

It is recommended to map fact tables and indices to one or two separate databases.

For the Fact and Indices databases journaling is optional and depends on the business needs. It might preferable to disable journaling when cubes are relatively small and fast to build, or cubes are scheduled to rebuild periodically.

Enable journaling on this database when cubes are relatively big and it takes too long to rebuild them. The ideal case to keep journaling on is when cubes are in a stable state and only get periodically synchronized, but not built. One way to safely build cubes is to temporarily disable journaling for the Fact database.

#Analytics #Beginner #Databases #Deployment #Mapping #Tutorial #IRIS Analytics (DeepSee)
50 4 0 5 487

Related posts

- DeepSee: Databases, Namespaces, and Mappings - Part 2 of 5
- DeepSee: Databases, Namespaces, and Mappings - Part 3 of 5
- DeepSee: Databases, Namespaces, and Mappings - Part 4 of 5

Log in or sign up to continue
Add reply

Source URL: https://community.intersystems.com/post/deepsee-databases-namespaces-and-mappings-part-3-5