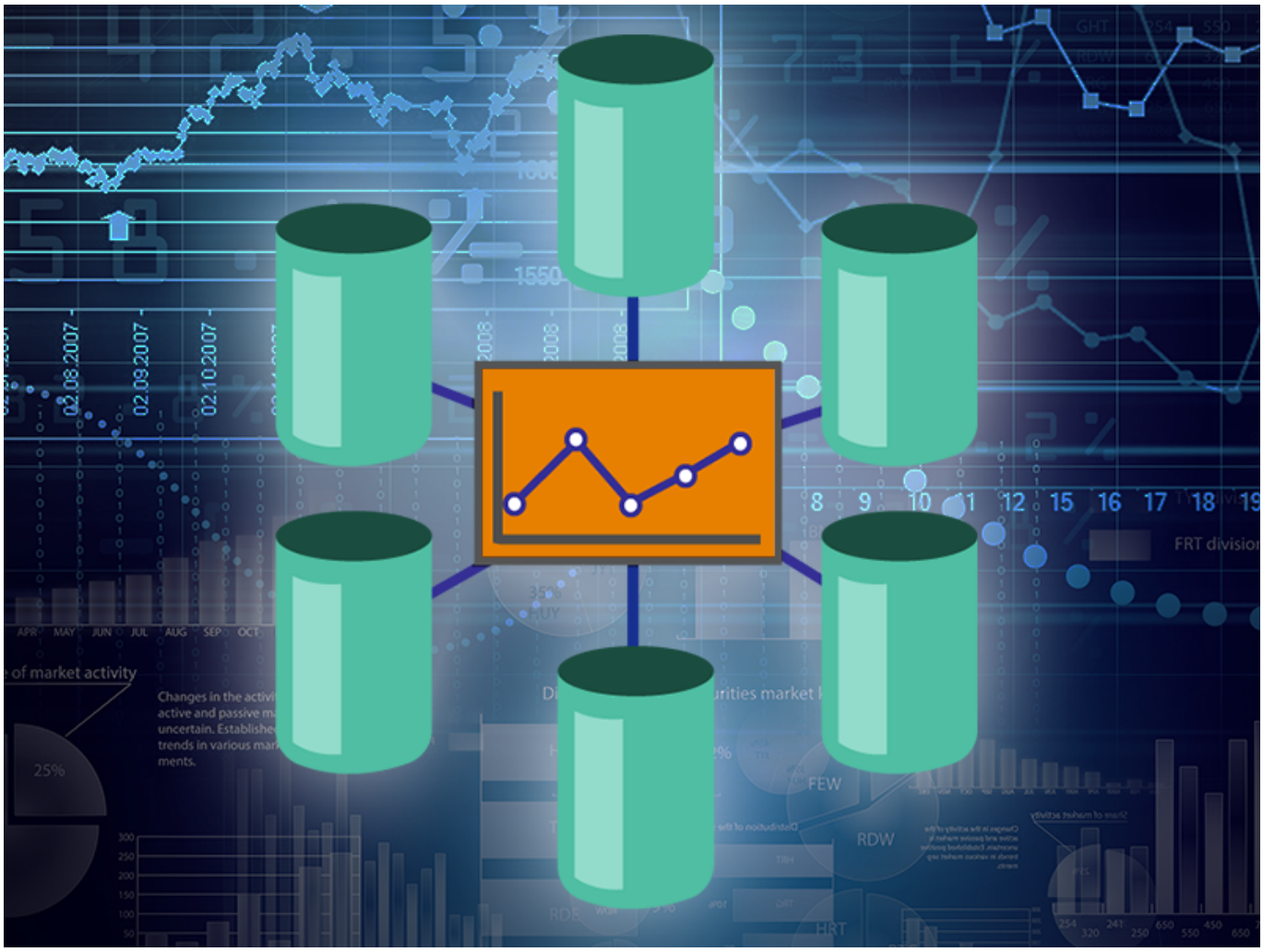
Artículo
Javier Lorenzo Mesa · 29 mar, 2021 Lectura de 3 min

DeepSee: Bases de datos, Namespaces y Mapeos - Parte 2 de 5

Esta publicación es una guía cuya finalidad es implementar una arquitectura básica para DeepSee. Esta implementación incluye una base de datos para la memoria caché de DeepSee y otra base de datos para la configuración e implementación de DeepSee.



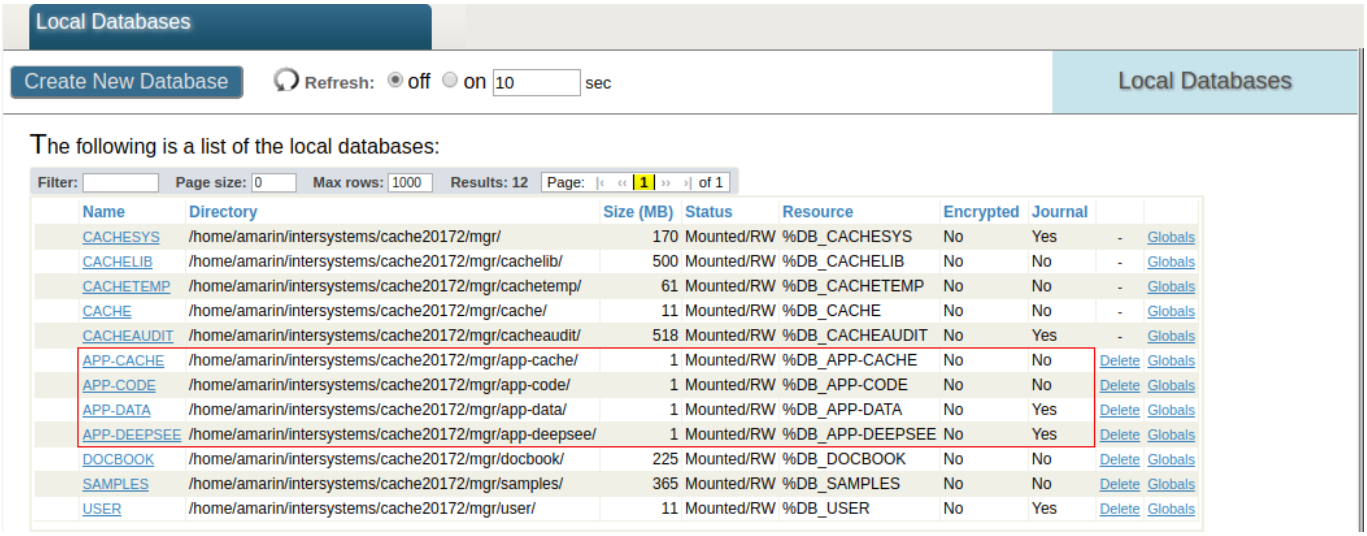
Ejemplo 1: Arquitectura básica

Bases de datos

En esta configuración el servidor (Analytics) incluye las bases de datos APP-CACHE y APP-DEEPSEE. Una configuración fundamental para que DeepSee se ejecute sin problemas es que **la memoria caché de DeepSee nunca debe ser registrada mediante journals**. Si se hace, disminuirá el rendimiento del motor de DeepSee, además de causar un hiper-registro y posiblemente se produzcan problemas de espacio en el disco. Por esta razón, la memoria caché de DeepSee se almacena en APP-CACHE, una base de datos separada del caché de DeepSee, con el registro deshabilitado.

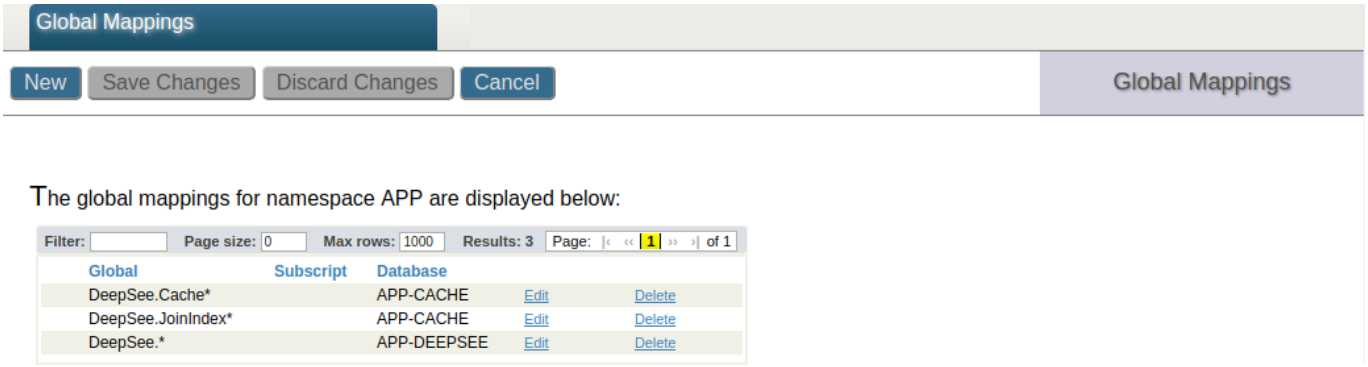
APP-DEEPSEE es una base de datos para la configuración e implementación de DeepSee, y que además

contiene los *globals* de ^DeepSee*. Estos *globals* definen la mayor parte de la implementación de DeepSee, así como las definiciones y configuraciones de los cubos, el Cube Manager, las configuraciones del usuario y muchas cosas más. Ten en cuenta, como verás en la captura de pantalla siguiente, que todas las bases de datos son de lectura y escritura y que se decidió habilitar el registro solamente en APP-DEEPSEE. Es recomendable registrar esta base de datos, ya que contiene todas las definiciones, configuraciones y datos de los usuarios. !



Mapeo de *globals*

La siguiente captura de pantalla muestra los mapeos de esta implementación arquitectónica básica en el namespace APP. ^DeepSee.Cache.* y ^DeepSee.JoinIndex mapean el caché de DeepSee hasta la base de datos APP-CACHE. Los *globals* de ^DeepSee.* mapean, entre otras cosas, la implementación de DeepSee y la configuración hacia la base de datos APP-DEEPSEE. !



Comentarios

En este ejemplo para una arquitectura básica, el caché de DeepSee se almacenó en una base de datos dedicada. Esto permite deshabilitar el registro en los *globals* ^ DeepSee.Cache * y ^DeepSee.JoinIndex.

El registro de la base de datos APP-DEEPSEE permite restaurar la implementación de DeepSee (cubos, áreas temáticas, elementos de DeepSee, configuraciones de usuario, etc.) en caso de que haya un evento desestabilizador.

La configuración que se describe en este ejemplo tiene algunos defectos. En primer lugar, no se tienen en cuenta los *globals* que participan en la sincronización. En segundo lugar, la base de datos APP-DEEPSEE también contiene tablas fact, índices y otros *globals* de DeepSee. De este modo, APP-DEEPSEE podría llegar a ser grande en tamaño, por lo que no sería práctico registrarla y restaurarla. Esta configuración podría considerarse

aceptable si, por ejemplo, los cubos no contienen una gran cantidad de datos.

En el siguiente ejemplo de esta serie, veremos cómo mapear *globals* de sincronización de cubos, tablas fact e índices para separar bases de datos.

#Analítica #Bases de datos #Despliegue #Mapeo #Principiante #Tutorial #InterSystems IRIS BI (DeepSee)

URL de fuente:https://es.community.intersystems.com/post/deepsee-bases-de-datos-namespaces-y-mapeos-parte-2-de-5