

Article

[Guillaume Rongier](#) · Avr 12, 2022 7m de lecture

## Structure interne des blocs de la base de données Caché, partie 1

Les globales d'InterSystems Caché offrent des fonctionnalités très pratiques pour les développeurs. Mais pourquoi les globales sont-elles si rapides et efficaces ?

### Théorie

Fondamentalement, la base de données Caché est un catalogue portant le même nom que la base de données et contenant le fichier CACHE.DAT. Sur les systèmes Unix, la base de données peut également être une partition de disque ordinaire.

Toutes les données dans Caché sont stockées dans des blocs qui, à leur tour, sont organisés sous forme d'un arbre B\* équilibré. En tenant compte du fait que tous les globales sont fondamentalement stockés dans un arbre, les indices des globales seront représentés comme des branches, tandis que les valeurs des indices des globales seront stockées comme des feuilles. La différence entre un arbre B\* équilibré et un arbre B ordinaire est que ses branches ont également des liens corrects qui peuvent aider à itérer à travers les souscripts (c'est-à-dire les globales dans notre cas) en utilisant rapidement les fonctions [\\$Order](#) et [\\$Query](#) sans revenir au tronc de l'arbre.

Par défaut, chaque bloc du fichier de la base de données a une taille fixe de 8 192 octets. Vous ne pouvez pas modifier la taille du bloc pour une base de données déjà existante. Lorsque vous créez une nouvelle base de données, vous pouvez choisir des blocs de 16 Ko, 32 Ko ou même 64 Ko, en fonction du type de données que vous allez stocker. Cependant, gardez toujours à l'esprit que toutes les données sont lues bloc par bloc - en d'autres termes, même si vous demandez une valeur unique d'un octet, le système lira plusieurs blocs parmi lesquels le bloc de données demandé sera le dernier. Vous devez également vous rappeler que Caché utilise des buffers globaux pour stocker les blocs de base de données en mémoire pour une seconde utilisation, et que les buffers ont la même taille que les blocs. Vous ne pouvez pas monter une base de données existante ou en créer une nouvelle si un buffer global avec la taille de bloc correspondante est absent du système. Vous devez définir la taille de la mémoire que vous souhaitez allouer pour la taille spécifique des blocs. Il est possible d'utiliser des blocs de buffer plus grands que les blocs de base de données, mais dans ce cas, chaque bloc de buffer ne stockera qu'un seul bloc de base de données, voire plus petit.

Menu [Home](#) | [About](#) | [Help](#) | [Logout](#) [System](#) > [Configuration](#) > [Startup Settings](#) > [Edit Startup Settings](#)

**Edit: DBSizesAllowed** Server: **MBP-Dmitry** Namespace: **%SYS**  
User: **UnknownUser** Licensed to: **2016.x Field Test** Instance: **CACHE**

[Save](#) [Cancel](#)

Use the form below to edit a startup setting:

**DBSizesAllowed** (Check all that apply. 8K is required.)

**8K (8192)**

**16K (16384)**

**32K (32768)**

**64K (65536)**

NOTE: You must also configure Memory allocation in order to create databases with selected sizes.

Stores a list of allowed database block sizes.

Dans cette image, une mémoire pour le buffer global d'une taille de 8 ko est allouée pour l'utilisation avec des bases de données constituées de blocs de 8 ko. Les blocs qui ne sont pas vides dans cette base de données sont définis dans des cartes, de sorte qu'une des cartes définit 62 464 blocs (pour des blocs de 8 ko).

## Types de blocs

Le système prend en charge plusieurs types de blocs. À chaque niveau, les liens corrects d'un bloc doivent pointer vers un bloc du même type ou vers un bloc nul qui définit la fin des données.

- Type 9: Le système prend en charge plusieurs types de blocs. À chaque niveau, les liens corrects d'un bloc doivent pointer vers un bloc du même type ou vers un bloc nul qui définit la fin des données.
- Type 66: Bloc de pointeurs de haut niveau. Seul un bloc d'un catalogue global peut se trouver au-dessus de ces blocs.
- Type 6: Bloc de pointeurs de bas niveau. Seuls les blocs de pointeurs de haut niveau peuvent se trouver au-dessus de ces blocs et seuls les blocs de données peuvent être placés plus bas.
- Type 70: Bloc de pointeurs de haut niveau et de bas niveau. Ces blocs sont utilisés lorsque la globale correspondante stocke un petit nombre de valeurs et que plusieurs niveaux de blocs ne sont donc pas nécessaires. Ces blocs pointent généralement vers des blocs de données, tout comme le font les blocs d'un catalogue global.
- Type 2: Bloc de pointeurs pour le stockage de globales relativement grandes. Afin de répartir uniformément les valeurs entre les blocs de données, vous pouvez créer des niveaux supplémentaires de blocs de pointeurs. Ces blocs sont généralement placés entre les blocs de pointeurs.
- Type 8: Bloc de données. Ces blocs stockent généralement les valeurs de plusieurs nœuds globaux plutôt que celles d'un seul nœud.
- Type 24: Bloc pour les grandes chaînes de caractères. Lorsque la valeur d'une seule globale est plus grande qu'un bloc, cette valeur est enregistrée dans un bloc spécial pour les grandes chaînes de caractères, tandis que le nœud de bloc de données stocke les liens vers la liste des blocs pour les grandes chaînes de caractères ainsi que la longueur totale de cette valeur.
- Type 16: Bloc de carte. Ces blocs sont conçus pour stocker des informations sur les blocs non alloués.

Ainsi, le premier bloc d'une base de données Caché typique contient des informations de service sur le fichier de base de données lui-même, tandis que le deuxième bloc fournit une carte des blocs. Le premier bloc de catalogue va sur la troisième place (bloc #3), et une seule base de données peut avoir plusieurs blocs de catalogue. Les blocs suivants sont des blocs de pointeurs (branches), des blocs de données (feuilles) et des blocs de grandes chaînes de caractères. Comme je l'ai mentionné ci-dessus, les blocs de catalogues globaux stockent des informations sur tous les globales existants dans la base de données ou les paramètres globaux (si aucune donnée n'est disponible dans une telle globale). Dans ce cas, un nœud qui décrit une telle globale aura un pointeur inférieur nul. Vous pouvez consulter la liste des globales existantes à partir du catalogue de globales sur le portail de gestion. Ce portail vous permet également de sauvegarder une globale dans le catalogue après sa suppression (par exemple, sauvegarder sa séquence de collationnement) ainsi que de créer une nouvelle globale avec un collationnement par défaut ou personnalisé.

En général, l'arbre des blocs peut être représenté comme dans l'image ci-dessous. Notez que les liens vers les blocs sont représentés en rouge.

## Intégrité des bases de données

Dans la version actuelle de Caché, nous avons résolu les questions et problèmes les plus importants concernant les bases de données, de sorte que les risques de dégradation des bases de données sont extrêmement faibles. Cependant, nous vous recommandons toujours d'exécuter régulièrement des contrôles d'intégrité automatiques à l'aide de notre outil ^Integrity - vous pouvez le lancer dans le terminal à partir de l'espace de noms %SYS, via notre portail de gestion, sur la page Database ou via le gestionnaire de tâches. Par défaut, le contrôle d'intégrité automatique est déjà configuré et prédéfini, de sorte que la seule chose que vous devez faire est de l'activer :

---

Le contrôle d'intégrité comprend la vérification des liens aux niveaux inférieurs, la validation des types de blocs,

l'analyse des bons liens et la mise en correspondance des nœuds globaux avec la séquence de collationnement appliquée. Si des erreurs sont détectées lors du contrôle d'intégrité, vous pouvez exécuter notre outil ^REPAIR à partir de l'espace de noms %SYS. Grâce à cet outil, vous pouvez visualiser n'importe quel bloc et le modifier si nécessaire, c'est-à-dire réparer votre base de données.

## Pratique

Cependant, ce n'était que de la théorie. Il est encore difficile de savoir à quoi ressemblent réellement une globale et ses blocs. Actuellement, la seule façon de visualiser les blocs est d'utiliser notre outil ^REPAIR mentionné ci-dessus. La sortie typique de ce programme est présentée ci-dessous :

Il y a un an, j'ai lancé un nouveau projet visant à développer un outil qui itère à travers un arbre de blocs sans risque d'endommager la base de données, qui visualise ces blocs dans une interface utilisateur Web et qui offre des options pour sauvegarder leur visualisation en SVG ou PNG. Le projet s'appelle CacheBlocksExplorer, et vous pouvez télécharger son code source sur [Github](#).

Les fonctionnalités mises en œuvre comprennent :

- Visualisation de toute base de données configurée ou simplement installée dans le système ;
- Affichage des informations par bloc, type de bloc, pointeur droit, liste des nœuds avec liens ;
- Afficher des informations détaillées sur tout nœud pointant vers un bloc inférieur ;
- Masquer des blocs en supprimant les liens vers ceux-ci (sans aucun dommage pour les données stockées dans ces blocs).

La liste des choses à faire :

Affichage des liens droits : dans la version actuelle, les liens droits sont affichés dans les informations sur les blocs, mais il serait préférable de les afficher sous forme de flèches ;

Affichage des blocs de grandes chaînes de caractères : ils ne sont tout simplement pas affichés dans la version actuelle ;

Affichage de tous les blocs du catalogue global plutôt que du troisième seulement.  
Je voudrais également afficher l'intégralité de l'arbre, mais je ne trouve toujours pas de bibliothèque capable de rendre rapidement des centaines de milliers de blocs avec leurs liens - la bibliothèque actuelle les rend dans les navigateurs web bien plus lentement que Caché ne lit la structure entière.

Dans mon prochain article, j'essaierai de décrire plus en détail son fonctionnement, de fournir quelques exemples d'utilisation et de montrer comment récupérer de nombreuses données exploitables sur les globales et les blocs à l'aide de mon Cache Block Explorer.

[#Administration du système](#) [#Bases de données](#) [#Caché](#)

---

URL de la source: <https://fr.community.intersystems.com/post/structure-interne-des-blocs-de-la-base-de-donn%C3%A9es-cach%C3%A9-partie-1>