Using the Caché ^GBLOCKCOPY Routine

Version 2009.1
30 June 2009

InterSystems Corporation   1 Memorial Drive   Cambridge MA 02142   www.intersystems.com

Using the Caché ^GBLOCKCOPY Routine
Caché Version 2009.1 30 June 2009
Copyright © 2009 InterSystems Corporation
All rights reserved.

This book was assembled and formatted in Adobe Page Description Format (PDF) using tools and information from the following sources: Sun Microsystems, RenderX, Inc., Adobe Systems, and the World Wide Web Consortium at www.w3c.org. The primary document development tools were special-purpose XML-processing applications built by InterSystems using Caché and Java.

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Customer Support

Tel:	+1 617 621-0700
Fax:	+1 617 374-9391
Email:	support@InterSystems.com

Table of Contents

Using the Caché ^GBLOCKCOPY Routine

This article describes the uses and procedures for running the Caché **^GBLOCKCOPY** routine to copy globals. It contains the following sections:

1 Overview

^GBLOCKCOPY is a Caché routine which performs fast global copies between databases. It can be run interactively to a terminal, or be set up in a batch to run one or more global copies as background jobs. **^GBLOCKCOPY** contains a built-in monitor and several reports to track the progress of global copies. You can restart **^GBLOCKCOPY** at the point it left off if there is a system failure.

Note: The **^GBLOCKCOPY** routine should only be used on globals that are currently not being accessed with **Set** or **Kill** operations. Results in the destination global are unpredictable if **Sets** or **Kills** occur in the source global which is being copied to another database or namespace. **Set** or **Kill** operations may be performed on other globals in the source database where the copy is being performed. They may also be performed in the destination global, database, or namespace without affecting the copy.

When **^GBLOCKCOPY** copies a global to a new database, it creates the global there with the same properties of the source global, including Protection, Journal attributes, Collation type, and Keep attributes. The one exception is if the global has one of the following collations:

- ISM Pre-6.2
- ISM 6.2->6.4
- Ipsum/Cobra
- DTM-compatible

- Ipsum/Cobra-2

In these cases, the routine automatically changes the collation to Caché Standard.

2 Uses of ^GBLOCKCOPY

^GBLOCKCOPY can be used for several different operations as follows:

- *Copy single or multiple globals from a database to another database or namespace* — You can select one or several globals to be copied into a destination database or namespace. If the global already exists in the destination database, data from the source global is merged into the existing data.
- *Convert a 2-kB database to a 8-kB database* — You can use the Auto Create Batch option to set up a batch of 2-kB to 8-kB conversions, or you may perform them one at a time as space and time permits.
- *Combine multivolume databases into a single volume* — If you have a database with several 2-GB volumes (common on UNIX platforms), you can select all the globals in this database and copy them into a single volume. Since databases can now exceed 2 GB in size, this is a useful method for consolidating several volumes into one database.
- *Split a global from a single database into multiple databases using subscript level mapping* — By setting up a namespace with subscript level mapping (SLM) of a global, you can copy a global from a database into this new namespace and cause it to be split amongst the database which make up SLM.
- *Move a subscript-mapped global in many databases into one database* — Create a new database which contains the entire global. Then set up several copies in a batch which will copy the global from all the different SLM databases into the new database. Since databases can now exceed 2 GB in size, this is a useful method for consolidating databases.
- Make a copy of a database — You may copy a database to another directory by copying all the globals to it.
- Copy a global to another machine across ECP — **^GBLOCKCOPY** supports copying a global across an ECP network connection to another machine. You need to set up an ECP connection to a remote machine, and a namespace mapping which points to it. Then select the “Copy from Database to Namespace” option and select the remote namespace as the destination of the copy.
- Reclaim unused space in a database — If a large global is created then killed in a database, there may be a large excess of unused space in the database. You can remove this space by copying all the globals in the database to a new one, and then replacing the old database with the new database.
- Reorganize the pointers in a database — If a database becomes fragmented because of block splits, you may want to reorganize the data in it to speed performance. You can do this by copying all

the globals in a database into a new database, and then replacing the old database with the new database.

- *Change collation of a global* — If a source global is of any of the following collation types:
  - ISM Pre-6.2
  - ISM 6.2->6.4
  - Ipsum/Cobra
  - DTM-compatible
  - Ipsum/Cobra-2

  The collation of the global is automatically changed to Caché Standard if it is copied via **^GBLOCKCOPY**.

  If you have an existing global which you want to change the collation of (for example, from Caché Standard to German1), create the destination database with the correct desired default collation before running **^GBLOCKCOPY**.

  Note: Databases which migrated from ISM 5.10/6.4 will contain globals with ISM collations.

- Convert an OPENM.DAT database to a CACHE.DAT database — Use the “Auto Create Batch” option to perform this conversion after an upgrade from OpenM.
- Convert an ISM 5.10/6.4 MUMPS.DAT database to a CACHE.DAT database — Use the “Auto Create Batch” option to perform this conversion after an upgrade from ISM.
- Import a legacy database into a CACHE.DAT database or namespace — If you have a legacy database file which you want to import to a CACHE.DAT database or namespace, simply select the directory where it exists as the source directory as the copy. The database is renamed CACHE.DAT, and the data is available to copy to the destination database or namespace.
- Import a ISM 5.10/6.4 MUMPS.DAT database into a CACHE.DAT database or namespace — If you have a MUMPS.DAT file which you want to import to a CACHE.DAT database or namespace, simply select the directory where it exists as the source directory as the copy. The database is renamed CACHE.DAT, and the data is available to copy to the destination database or namespace.

3 Running ^GBLOCKCOPY

Before you run **^GBLOCKCOPY** (or for that matter before you perform an upgrade), make a full operating system backup of your databases, and run an integrity check to ensure there is no corruption in any of the databases.

Note: To make **^GBLOCKCOPY** run faster, kill off any temporary and scratch data as well as any old data you do not require. For ISM upgrades, do this before you perform the upgrade. For Caché 2-kB to 8-kB conversions, you can do this just before you run **^GBLOCKCOPY**.

You can use the batch functionality of **^GBLOCKCOPY** to set up a batch of conversions to run at the same time. Select the “Auto Create a Batch” option. This option looks through the *^SYS("UCI")* global for a list of the databases on the system. If it detects that any of the databases are using a 2-kB size then it gives the option to add it to the batch of conversions. Once the batch of conversions is set up, you can then select the option to run the batch. While the batch of conversions is running, you can monitor progress using the Monitor or Batch Report.

Note: Users should be kept from accessing the databases while they are being converted with **^GBLOCKCOPY**. The result of the database conversions are unpredictable if they are accessed while **^GBLOCKCOPY** is running. Databases on the same system which are not being converted with **^GBLOCKCOPY** can be used safely.

The following is an example of using the “Auto Create a Batch” option:

```
%SYS>d ^GBLOCKCOPY

This routine will do a fast global copy from a database to another database or
to a namespace. If a namespace is the destination, the global will follow any
mappings set up for the namespace.


1) Interactive copy
2) Batch copy
3) Exit

Option? 2

1) Manage Batches
2) Run a Batch
3) Restart a Batch
4) Add Processes to a running Batch
5) Stop a Running batch
6) Monitor Running Batch
7) Batch Report
8) Exit

Option? 1

1) Create a Batch
2) Edit a Batch
3) List Batches
4) Delete a Batch
5) Auto create a Batch
6) Exit

Option? 5

1) Convert 2K DB to 8K DB
2) Convert OPENM/ISM DB
3) Both
4) Exit

Option? 1

Batch name to auto create? CVT2K
```

```
Detected 2K c:\temp\a\CACHE.DAT, convert to 8K CACHE.DAT? Yes => Yes
Detected 2K c:\temp\x\CACHE.DAT, convert to 8K CACHE.DAT? Yes => Yes
Detected 2K c:\temp\z\CACHE.DAT, convert to 8K CACHE.DAT? Yes => No

Auto creating the following database conversions:

Convert 2K c:\temp\a\CACHE.DAT => 8K c:\temp\a\CACHE.DAT
Convert 2K c:\temp\x\CACHE.DAT => 8K c:\temp\x\CACHE.DAT

Confirm auto creation of batch CVT2K? No => yes

Database c:\temp\a\CACHE.DAT renamed to c:\temp\a\old\CACHE.DAT
Database c:\temp\x\CACHE.DAT renamed to c:\temp\x\old\CACHE.DAT

Batch Name: CVT2K  0 of 0 Processes running, 0 per directory
 # Source DB=>Destination DB/NS                  Last Update Done/ToDo/Err State
-- ----------------------------                  ----------- ------------ -----
 1 c:\temp\a\old\=>c:\temp\a\                                  0/  15/  0 Queue
 2 c:\temp\x\old\=>c:\temp\x\                                  0/  15/  0 Queue

Auto creation of batch CVT2K complete

1) Convert 2K DB to 8K DB
2) Convert OPENM/ISM DB
3) Both
4) Exit

Option? 4

1) Create a Batch
2) Edit a Batch
3) List Batches
4) Delete a Batch
5) Auto create a Batch
6) Exit

Option? 6

1) Manage Batches
2) Run a Batch
3) Restart a Batch
4) Add Processes to a running Batch
5) Stop a Running batch
6) Monitor Running Batch
7) Batch Report
8) Exit

Option? 2

Batch name to run? ?

Name    #  Source DB=>Destination DB/NS                       Globals  Status
----    -  ----------------------------                       -------  ------
CVT2K   1  c:\temp\a\old\=>c:\temp\a\                         All      Queue
        2  c:\temp\x\old\=>c:\temp\x\                         All      Queue


Batch name to run? CVT2K
How many copy processes do you want to run at once? 2 =>
How many of these copy processes do you want to run for each directory? 1 =>
Confirm copy? Yes =>
```

4 Performing Upgrade Tasks with ^GBLOCKCOPY

You can use the **^GBLOCKCOPY** routine to perform several tasks associated with upgrading as described in the following sections:

- Using ^GBLOCKCOPY to Convert Databases
- Converting the Manager's Database
- Differences between %SYS.SYSCONV and ^GBLOCKCOPY

4.1 Using ^GBLOCKCOPY to Convert Databases

After you have performed the upgrade to Caché, you can use the **^GBLOCKCOPY** routine to perform the following operations:

- Convert ISM 5.10 and 6.4 MUMPS.DAT databases to CACHE.DAT databases instead of using the **%SYS.SYSCONV** routine.
- Convert other legacy databases to CACHE.DAT databases instead of using the **%SYS.SYSCONV** routine.
- Convert any 2-kB databases to 8-kB databases.

Note: InterSystems recommends that you convert existing 2-kB databases to 8-kB databases for improved performance. This conversion can happen either at the time of the upgrade, or anytime afterwards. You can convert databases one or several at a time as down time and disk space permit.

4.2 Converting the Manager's Database

After an upgrade from older versions of Caché, the manager's database is left at a 2-kB size. You can use **^GBLOCKCOPY** to convert this to an 8-kB as follows:

1. Verify that no users are on the system.
2. Run **^GBLOCKCOPY** to copy the manager's database to a different directory.
3. Shut down Caché cleanly.
4. Copy the newly created database into the manager's directory.
5. Restart Caché.

The following is an example:

```
%SYS>d ^GBLOCKCOPY

This routine will do a fast global copy from a database to another database or
to a namespace. If a namespace is the destination, the global will follow any
mappings set up for the namespace.

1) Interactive copy
2) Batch copy
3) Exit

Option? 1

1) Copy from Database to Database
2) Copy from Database to Namespace
3) Exit

Option? 1
Source Directory for Copy (? for List)? c:\cachesys\mgr
Destination Directory for Copy (? for List)? c:\temp\mgr
Database c:\temp\mgr\ does not exist
Do you want to create it? No => Yes
Database c:\temp\mgr\ created
All Globals? No => Yes

39 items selected from
39 available globals

Turn journaling off for this copy? Yes => Yes
Confirm copy? Yes => Yes
```

4.3 Differences between %SYS.SYSCONV and ^GBLOCKCOPY

The **%SYS.SYSCONV** routine renames the MUMPS.DAT databases to CACHE.DAT, performs an in-place conversion of globals which stores the routine source and object code (no additional disk space is required), and then recompiles the routines. The databases are still 2-kB size. The **%SYS.SYSCONV** routine is only used for ISM 5.10/6.4 upgrades. The **%SYS.SYSCONV** routine runs slowly since it performs the upgrade without using additional disk space, and only moves single nodes of data at a time.

The **^GBLOCKCOPY** routine requires enough free disk space to make a complete copy of the original database. While running the routine you enter information to rename the original database to CACHE.DAT in the *old* subdirectory of the original database. It creates a new 8-kB database in the database directory, then copies the data from the *old* directory into the new database. After the conversion, you can delete the *old* database after validating the converted data. The **^GBLOCKCOPY** routine runs quickly since it is not constrained to run in the same amount of disk space, and moves an entire block of data at a time. However, because it is moving *all* the data in the database rather than just the routine and object code (as **%SYS.SYSCONV** does), it may take more time to run than **%SYS.SYSCONV**, depending on the size of the database.

Note: As you convert databases from 2-kB to 8-kB, shift the number of MB allotted to the 2-kB database cache to the 8-kB database cache on the **[Home] > [Configuration] > [Memory and Startup]** page of the System Management Portal.