CHAPTER

10

Additional Global Utilities

This chapter describes additional global utilities that enable you to examine and manipulate routines from terminal devices. All of the functionality described in this chapter is also available through the Caché Explorer.

These utilities are provided for use in batch jobs or an alternative to use when you do not have a client machine handy.

Selecting Globals and Global Nodes

This section describes how to specify globals and global nodes for selection. You use the methods described here for both GUI and character-based utilities.

Specifying Global Names to Select Globals

This section describes methods for selecting globals. These responses will typically be to the Global ^ prompt.

Methods for Specifying Globals

The simplest way to specify globals is to enter the name of each global:

```
Global ^LAB
```

selects the global LAB.

You can also use the following symbols.

Symbol	Definition
? (question mark)	• A question mark by itself displays all available globals and flags those globals that have been selected. • A question mark followed by any character displays just the globals that have been selected. • A question mark within a specification matches any single character.
; (semicolon)	Separates the globals in a list of globals.
- (dash)	Indicates an inclusive range of globals to be selected.
' (apostrophe)	Indicates globals to be excluded from those to be selected.
* (asterisk)	A wildcard symbol that matches any character(s).
& (ampersand)	A wildcard symbol that matches any letter.
# (pound)	A wildcard symbol that matches any digit.

The asterisk can be placed after a set of alphanumeric characters to specify all globals whose names start with those characters. This example selects all globals whose name starts with R:

```
Global ^R*
```

The asterisk alone selects all globals:

```
Global ^*
```

Use a dash to select a range of globals. This example selects all globals whose names range from XR and YB inclusively:

```
Global ^XR-YB
```

Use an apostrophe to exclude one or more globals from those already selected. The following specification excludes the global NIM:

```
Global ^'NIM
```

You can combine the apostrophe with the dash or an asterisk to specify a set of globals to exclude. This example excludes all globals whose names begin with AB and whose names range from XR to YB inclusively:

```
Global ^'AB*
Global ^'XR-YB
```

To select all globals except a specified set, specify all globals and then the set to be excluded. This example selects all globals except the global ABC:

```
Global ^*
Global ^'ABC
```

Wildcard symbols can be combined in any order. The following examples illustrate how to combine wildcard symbols:

Global ^*A**	selects all globals whose names begin with A
Global ^**T**	selects all globals whose names end with T
Global ^**#**	selects all globals whose names end with a digit
Global ^**1**	selects all globals whose names contain the digit 1
Global ^**Z??**	selects all globals whose names begin with Z and are exactly three characters long

To see how many globals match a set of specifications, select them with %GSET. For example, the following call to %GSET reveals that 10 globals in the indicated namespace match the indicated specifications.

```
%SYS>D ^%GSET

All globals? (Y or N) N= <RETURN>

Global ^LAB
Global ^N*
Global ^'NIM
Global ^XR-YB
Global ^<RETURN>
10 globals
%SYS>
```

Selected globals are stored in the global ^UTILITY. To see what globals match a set of specifications, select them with %GSET and then use %G to examine the nodes of ^UTILITY.

The following call to %G shows the globals selected by the call to %GSET.

```
%SYS> D ^%GSET
All globals? (Y or N) N= Y

Global ^LAB
Global ^DI-DXA
Global ^B*
Global ^'BAC
Global ^'BX-BYA
Global ^'BIL*
Global ^<RETURN>
11 globals

%SYS> D ^%G

Global ^UTILITY($J#256,
^UTILITY(2,"BAA")=
      "BAB")=
          "BACC")=
         "BCODE")=
           "BWA")=
           "BYB")=
           "DIC")=
          "DICT")=
           "DLA")=
           "DXA")=
           "LAB")=
```

All the symbols described above can also be used with utilities that call %GSET. To display selected globals either alone or flagged as selected, together with all other globals, use the Caché Explorer.

Importing and Exporting Globals

Caché provides six utilities for importing and exporting globals for use on other systems and remote storage.

- The %GI utility imports globals into Caché. This is the most common method of importing whole globals.
- The %GO utility exports globals to disk or tape. This is the most common method of exporting whole globals.
- The %GIF utility imports globals to disk or tape. This method works faster than %GI but it doesn't allow you to import globals that span namespaces.
- The %GOF utility exports globals to disk or tape. This method works faster than %GO but doesn't allow you to export globals that span namespaces.
- The %GIGEN utility imports globals or selected portions of globals into Caché. It also allows you to restore a global subtree to a different global.
- The %GOGEN utility exports globals or selected portions of globals to disk or tape.

Importing Globals with %GI

The %GI utility is used to restore those globals previously saved with the %GO utility.

The %GI utility does the following:

- Calls the %IS utility to determine which device will be used as the input device.
- Reads and displays the first two strings from that device, which should be the date and time that the globals were saved and the description comment
- Asks you to select an input option. You can enter "A" to restore all globals, "S" to select globals to be restored individually, or press <RETURN> to restore nothing. As each global is loaded, its global name is displayed on the terminal.

This example shows a typical global import:

```
%SYS>D ^%GI

Global input

Device: c:\cachesys\global   Parameters: ("R")=><RETURN>

Globals were saved on Apr 16 99 2:07 PM
with description: Save globals for backup
Input option: ?
Select A to restore all globals or S to select globals to restore, one by one
Input option: S

Okay to restore ^ACCOUNT? No= Y
Okay to restore ^NAME? No= N
Okay to restore ^SPOOL? No= Y
%SYS>
```

Exporting Globals with %GO

The %GO utility is used to output a set of globals to media such as a disk or magnetic tape or to print a set of globals on a terminal. If the output is to a nonterminal device, the following format is used:

```
description (any user supplied text)
date and time
global reference
global data (output via WRITE command)
global reference
global data
null string (terminates the file)
```

The %GO utility does the following:

- Calls the device selection utility %IS to determine where output should be directed
- Asks for a comment to be used as a description
- Calls the %GSET utility to select a set of globals to be output
- Outputs the globals to the specified device in alphabetical order.

Each time a global is output, the terminal that started the output prints the global name.

Note that if the data in the globals to be saved contains control characters (such as form feeds or vertical tabs) and you are writing to magnetic tape, use the format AUV. This avoids problems in restoring the globals later.

The following is an example of using the %GO utility:

```
%SYS>D ^%GO

Global output
Device: c:\cachesys\global   Parameters: "WNS"=><RETURN>

Description (^ for self-loading): Save globals for backup
All globals? (Y or N) No= N

Global ^ACCOUNT
Global ^NAME
Global ^SPOOL
Global ^
3 globals written
%SYS>
```

Importing and Exporting Globals Quickly with %GIF and %GOF

When importing or exporting simple, whole globals, ones that do not span namespaces, you can use %GIF and %GOF for faster global import and export. They work identically to %GI and %GO.

Importing Part of a Global with %GIGEN

The %GIGEN utility is used to restore globals saved by %GOGEN. %GIGEN and %GOGEN are slower than %GI and %GO, but they offer a more general set of features, including the ability to restore a global subtree under a different global.

The %GIGEN utility does the following:

- Calls the %IS utility to determine which device will be used as the input device.
- Asks you to select an input option. You can enter "Yes" to restore all globals to the same global node as used when saving the global, or "No" to select globals to be restored individually to either the same global node or a different global node.

If the global specification used when %GOGEN saved a file was in the format,

```
^GLO("Y",2
```

or

```
^GLO("Y",2,
```

the utility will allow you to specify an alternative location, such as,

```
^X(1,2,3
```

In this example, the node ^GLO("Y",2) would be restored to the node ^X(1,2,3), and the node ^GLO("Y",2,7) would be restored as ^X(1,2,3,7).

Note that remapping of globals saved through more exotic specifications in %GOGEN, such as ^GLO(,2,3:4, is not allowed. In such cases, the restoration is to the same global reference used as input.

This example shows two %GIGEN examples then displays the global:

```
%SYS>D ^%GIGEN

Device: c:\cachesys\global   Parameters: "R"=> <RETURN>

Transfer entire set of files?  No=> <RETURN>
Transferring files on Apr 16 1999 at 9:17 AM
From global ^NAME("SMITH",,,
To global ^NAME("SMITH",,,

Transfer completed
Transferring files on Apr 16 1999 at 9:17 AM
From global ^NAME("WILSON",,
To global ^NAME("WILSON",,

Transfer completed
Done for this set of files.

%SYS>D ^%GIGEN

Device: 47  Parameters: ("AUV:0:2048)   Rewind? No= Y

Transfer entire set of files? No= <RETURN>
Transferring files on Apr 16 1999 at 9:17 AM
From global ^NAME("SMITH",,,
To global ^NAME("SMITH",,,

OK to transfer? Yes= <RETURN>

Transfer completed
Transferring files on Apr 16 1999 at 9:17 AM
From global ^NAME("WILSON",,
To global ^NAME("WILSON",,=^PERSON("CLIENT","WILSON",,

OK to transfer? Yes= <RETURN>

Transfer completed
Done for this set of files.
%SYS>D ^%G

Global ^NAME("WILSON",
```

```
^NAME("WILSON","HOWARD")=3003 5TH AVE
       "JANE")=55 MAIN ST
          "ROBERT")=404 ELM ST
Global ^PERSON
^PERSON("CLIENT","WILSON","HOWARD")=3003 5TH AVE
            "JANE")=55 MAIN ST
           "ROBERT")=404 ELM ST
Global ^
%SYS>
```

Exporting Part of a Global with %GOGEN

The %GOGEN utility can save selected portions of a global, unlike %GO and %GOF, which can save only an entire global at a time. Thus while %GO and %GOF are faster, %GOGEN offers the advantage of more general functionality.

In some cases, the %GIGEN utility, which restores global nodes saved with %GOGEN, can restore them under a different global subtree.

%GIGEN uses the following format to output to a nonterminal device:

```
null string
TRANSFERRING FILES ON date and time
user response to "GLOBAL ^" question (reformatted)
null string
global reference
data
global reference
data
global reference
data
DONE
```

The %GOGEN utility does the following:

- Calls the device selection utility %IS to determine where output should be directed.
- Calls the %GSET utility to select the globals or global subscripts to be output.
- Outputs the globals to the specified device in alphabetical order.

In the first two examples, the output device is specified as your terminal so that you can see the output. Normally, of course, you would specify a device such as sequential file or magnetic tape sequential file, as in the third example.

```
%SYS>D ^%GOGEN

Device: <RETURN>   Right margin: 80= <RETURN>

[Warning: Use a "V" format to avoid problems with control characters.]

Global ^NAME("WILSON",

Transferring files on Apr 16 1999 at 9:26 AM
^NAME("WILSON",

^NAME("WILSON","HOWARD")
3003 5TH AVE
^%NAME("WILSON","JANE")
55 MAIN ST
^NAME("WILSON","ROBERT")
404 ELM ST
DONE

Global ^

%SYS>D ^%GOGEN

Device: <RETURN> Right margin: 80= <RETURN>

[Warning: Use a "V" format to avoid problems with control characters.]

Global ^NAME("SMITH",
Transferring files on Apr 16 1999 at 9:28 AM
^NAME("SMITH",

^NAME("SMITH","NANCY","CITY")
MIAMI
^NAME("SMITH","ROBERT","CITY")
CHICAGO
DONE

Global ^

%SYS>D ^%GOGEN

Device: c:\cachesys\swglobal Parameters: ("WNS")=> <RETURN>
Global ^NAME("SMITH",)

Global ^NAME("WILSON",)

Global ^
%SYS>
```

Copying Globals

Use the %GCOPY utility to copy one global to another. This provides a quicker way to make a copy than exporting a global and importing it to a different name.

To copy one global to another:

1. Invoke %GCOPY. Caché prompts you for the global you want to copy:
   Copy Global ^
2. Enter the name of the global you want to copy. Caché prompts you for the namespace where the global is stored; default is the current namespace:
   Copy Global ^***zem***
   in namespace: %SYS =>
3. Enter the name of the namespace where the global is stored. Caché prompts you for the name of the target global; default is the source global.
4. Enter the name of the global to which you want to copy. Caché prompts you for the namespace where the target global will be stored; default is the current namespace:
   To Global ^zem=> ***^zeb***
   in namespace: %SYS =>

The global and all its contents are copied into a new global with the name and namespace you specified.

This example shows a global copy:

```
%SYS>set ^zem="Hello"

%SYS>d ^%GCOPY
 Copy Global ^zem
  in namespace: %SYS =>

 To Global ^zem=> ^zeb
  in namespace: %SYS =

%SYS>w ^zeb
Hello
```

Listing Globals and Attributes

Use %GD to get a list of the global in the selected namespace and details about the globals including:

- Access privileges for each category of user: owner, group, world, and network
- Data growth area
- First pointer block
- Whether journaling is set for the global
- Target directories for mapped and replicated globals

The data growth area and first pointer block fields help you locate globals within a database. The report also identifies hidden, mapped, and replicated globals and global keys (patterns). For mapped and hidden globals it shows no other information.

This example shows a list of the globals in the current namespace:

```
USER>DO ^%GD

Device:      Right margin: 80=>

Show detail? No => yes

                        Global Directory Display of USER
                               2:52 PM  Jul 30 99

Global          Vacant    Own Grp Wld Net  Growth    1st PB    Jrn

^%qCacheObjectCDL         RWD R   R   RWD  396       84        N
     Data Location:       c:\cachesys\mgr\cachelib\
     Lock Location:       c:\cachesys\mgr\cachelib\
     Replications:
     Collation:           Cache standard

^%qCacheObjectError       RWD R   R   RWD  396       85        N
     Data Location:       c:\cachesys\mgr\cachelib\
     Lock Location:       c:\cachesys\mgr\cachelib\
     Replications:
     Collation:           Cache standard

19 globals listed.
USER>
```

Viewing Global Nodes

The %G utility shows global nodes. %G first calls %IS to select a device for the displayed global node output, then asks for the global name. If the value of the variable %IO is already equal to the current device, the %G utility skips the "Device:" prompt.

The following sections describe methods for specifying global nodes to be selected for display in response to the Global ^ prompt displayed when you call %G. If you respond to the "Global ^" prompt with a question mark, %G displays the global directory.

Specifying a Global Subtree

If you enter the name of a global, %G displays the entire global. If you enter a complete global reference, such as "^GLO(3,"BED",5)", only that particular node is displayed. You may also specify a subtree, such as "^GLO(3,"BED"," in which case all descendants of that node are displayed. To display both the node and its descendants, do not end your entry with a comma or a right parenthesis.

This example displays a global subtree and all decendants of the node:

```
USER>D ^%G
Device: <RETURN>  Right Margin: 80=> <RETURN>
Global ^GLO(3,"BED"
^GLO(3,"BED")=EAST WING
^GLO(3,"BED",0)=123
          "ABC")=
          "CAT")=45
^GLO(3,"BED","TD",1)=MERRY CHRISTMAS
             34)=HAPPY NEW YEAR
Global ^
```

Quotation Marks

Use quotation marks when specifying a string subscript. Noncanonic numbers are also enclosed in quotation marks. (A canonic number is a number that satisfies the equation +X=X. A noncanonic number has superfluous zeros, or an explicit plus sign.) If a subscript includes a quotation mark as a subscript character, it appears as two quotation marks in the subscript. Quotation marks do not enclose the data value, however, and a quotation mark in the data appears as only a single quote in the data value. Thus, with a null string, there are no characters to the right of the equal sign.

Pointer Only Nodes

If a node is displayed whose $D() value is ten (a pointer with no data), then the word "pointer" appears after the global reference without any equal sign. For example:

```
^ABC(3,10)pointer
^ABC(3,10,2)="Johnson,Johnny"
```

Null Subscripts in Specification

You can leave a subscript field empty when you specify the subtree for display. The %G utility displays any nodes matching the other subscripts and having any value for the missing subscript. For example:

```
Global ^SCRATCH(,8)
^SCRATCH(3,8)=12
^SCRATCH(5,8)=333
^SCRATCH(12,8)=1234
Global ^
```

In this example, the %G utility displayed all the nodes in ^SCRATCH whose second level subscript was 8.

Use of a Right Parenthesis

In the previous example, a right parenthesis appeared in the specification, so %G displayed no nodes with more than the specified two subscript levels. If a right parenthesis is not present, the utility also displays any descendants of these nodes, as in the following example:

```
Global ^SCRATCH(,8
^SCRATCH(3,8)=12
^SCRATCH(3,8,1)=4
              6)=23
^SCRATCH(3,8,6,12)=1
^SCRATCH(5,8)=333
^SCRATCH(5,8,1,3)=4
^SCRATCH(12,8)=1234
Global ^
```

Subscript Ranges

You can also specify a range of subscripts for a particular subscript level by inserting a colon between the first and last subscript in the range:

```
Global ^PT(1,"ACC":"BIRTH"
^PT(1,"ACC")=123456
      "ADD")=1 PETER STREET
    "BIRTH")=3/12/47
Global ^
```

If no subscript is in front of the colon, the utility starts with the first subscript present at that level. If there is no subscript after the colon, the utility ends with the last subscript present at that level.

Using Variables and Expressions

You can also use variables and simple expressions in a subscript specification. For example, if you previously assigned a value of 514 to the variable "ID", using %G would result in:

```
Global ^PT(ID,
^PT(514,0)=JONES,JIMMY^1234^456
        1)=31456
        5)=45645
Global ^
```

The %G utility uses variables that start with a percent sign for its working variables. In subscript specifications you can use variables present in the symbol table when the %G utility was called. You can also use Caché expressions that do not include global variables.

Any Combination Works

You can combine any of the foregoing capabilities. For example:

```
Global ^GLO(,23:67,X,
```

This example shows the contents of the global in the %SYS namespace:

```
%SYS>D ^%G

Device: <RETURN> Right margin: 80= <RETURN>

Global ^?
        Global Directory Display of %SYS
           2:10 PM  Nov 24 99

^%                ^%CDServer         ^%CDUaf           ^%IS
^%RS              ^%SYS              ^%nls             ^%tercap
^%utility         ^CacheTemp         ^NET              ^ROUTINE
^SYS              ^rINC              ^rOBJ             ^NAME

16 globals listed.

Global ^NAME
^NAME("CABOT","THOMAS")=133 ELM ST
^NAME("CARLSON","JAMES"=444 PINE ST
^NAME("JOHNSON","BARBARA")=202 NORTH ST
^NAME("JONES","LINDA")=188 PINE ST

^NAME("SMITH","NANCY")=155 WASHINGTON ST
^NAME("SMITH","NANCY","CITY")=MIAMI
^NAME("WILSON","HOWARD")=3033 5TH AVE
                 "JANE")=55 MAIN ST
               "ROBERT")=404 ELM ST

Global ^NAME("CARLSON","JAMES")
^NAME("CARLSON","JAMES")=444 PINE ST

Global ^NAME("SMITH")
^NAME("SMITH")pointer

Global ^NAME("SMITH",
^NAME("SMITH","NANCY")=155 WASHINGTON ST
^NAME("SMITH","NANCY","CITY")=MIAMI

Global ^NAME("SMITH",)
^NAME("SMITH","NANCY")=155 WASHINGTON ST
         "NANCY")pointer

Global ^NAME("WILSON","H":"M")
^NAME("WILSON","HOWARD")=3003 5TH AVE
        "JANE")=55 MAIN ST

Global ^<RETURN>
%SYS>
```

Modifying Global Nodes

Use the %GCHANGE utility to search a global for all occurrences of a specified sequence of characters and:

- Display the nodes and their values
- Change their values by substituting one sequence of characters for another. As values are modified, the utility displays the global nodes and their new values.
- Execute a specified line of code, using the XECUTE command, for each data field that matches a specified sequence of characters. After the execute is completed, the utility prints each located data field.

To use %GCHANGE display nodes, change values, or execute code:

1. Invoke %GCHANGE. The utility uses the %IS utility to select a device for writing the output. Then it asks you to specify a global by asking:

   ```
   Global ^
   ```

2. Answer with any of the formats used for the %G global display utility.
3. Enter the function you want when the utility asks which function you want to perform:

   ```
   (F) Find, (C) Change every, or (X) Execute code:
   ```

This example shows a global find, and then a global change:

```
%SYS>D ^%GCHANGE
Output results on
Device: <RETURN> Right margin: 80=><RETURN>

Global ^A(5:200,3,
(F) Find, (C) Change every, or (X) Execute code: F=><RETURN>
Find every:100
^A(6,3,0)=100
^A(19,3,12,40)=41003

Global ^A(,4,
(F) Find, (C) Change every, or (X) Execute code: F=> C
Change every: X to ZA
^A(3,4,5)="ABZAM"
^A("SMITH",4,2,0)="WZAYZ"
Global ^
%SYS>
```

If you select X for Execute, the utility asks for a line of code to be executed every time it finds a matching node. If you wish, you can specify a string of characters that must be present in the data for the code to be executed. You may answer the "Find every data node with:" question with a null response. When the utility executes your code, the variable %Q holds the full name of the node found, and

%R contains its value. You can use these values in your code. In any event, the utility displays the name and value of each node after it executes your code.

Caution: The %GCHANGE utility uses % variables and IO variables which your XECUTE code should not change. If it does, the results are unpredictable. However, you can use indirection (as in the example below) to change the value of the node found.

This example shows a global XECUTE on a data node:

```
%SYS>D ^%GCHANGE
Output results on
Device: <RETURN> Right margin: 80=> <RETURN>
Global ^A
(F) Find, (C) Change every, or (X) Execute code: F=> X
%Q is the full global reference, %R its value.
Do NOT change any % variables!
Find every data node with: 100
And execute code: W !,"VALUE OF ",%Q," WAS ",%R,! S @%Q=%R+4
VALUE OF ^A WAS 100
^A=104
VALUE OF ^A(2,40) WAS 4100
^A(2,40)=4104
Global ^
%SYS>
```