Article
Cindy Olsen · Nov 8, 2016 7m read

SYSLOG - what it really is and what it means

In this post I would like to talk about the syslog table. I will cover what it is, how you look at it, what the entries really are, and why it may be important to you. The syslog table can contain important diagnostic information. If your system is having any problems, it is important to understand how to look at this table and what information is contained there.

What is the syslog table?

Caché sets aside a small portion of its shared memory to log items of interest. This table is referred to by several different names:

- Caché System error log
- errlog
- SYSLOG
- syslog table

For the purposes of this post, I ' m going to simply call it the ' syslog table ' .

The size of the syslog table is configurable. The default is 500 entries. The range is 10 to 10,000 entries. To change the size of the syslog table, use the System Management Portal -> System Administration -> Configuration -> Additional Settings -> Advanced Memory, in the ' errlog ' row, select ' edit ' . Enter the number of entries you would like in your syslog table.

Why would I want to change the size of my syslog table?

If the syslog table is configured for 500 entries, the 501st entry will overwrite the first entry, and that information will be lost. This is a table in memory, and therefore does not persist anywhere, unless you specifically save the output. Also, when you stop Caché, all of the entries will be lost, unless you have it configured to save the entries to the cconsole.log file, as described below.

If Caché is putting a lot of entries into the syslog table, and you want to look at them in order to help diagnose any problems, you will lose the entries if the table is not big enough. You can look at the **Date/Time** column in the syslog table to determine the time period it takes to fill the table. You can then decide how many entries you would like. I like to err on the side of not losing any entries. This will be described in greater detail below.

How do I look at the syslog table?

There are several ways to look at the syslog table:

1. From a Caché terminal prompt in the %SYS namespace, ' Do ^SYSLOG '
2. From a Caché terminal prompt in the %SYS namespace, ' do ^Buttons '
3. Management Portal -> System Operation -> Diagnostic Report
4. Run cstat with the –e1 option
5. Run Cachehung
6. Configure Caché to log the syslog table to the cconsole.log file during shutdown, and look at the cconsole.log file. To do this, use the System Management Portal -> Configuration -> Additional Settings ->

Compatibility, in the ‘ ShutDownLogErrors ’ row, select ‘ edit ’. Choose ‘ true ’ to save the syslog contents to cconsole.log at Caché shutdown, or ‘ false ’ to not save them

What are the syslog entries?

Below is an example of the syslog table. This example came from running ^SYSLOG at the Caché terminal prompt:

```
%SYS>d ^SYSLOG

Device:
Right margin: 80 =>
Show detail? No => No
Cache System Error Log printed on Nov 02 2016 at  4:29 PM
------------------------------------------------------
Printing the last 8 entries out of 8 total occurrences.
Err  Process         Date/Time                          Mod     Line      Routine
                           Namespace
9     41681038     11/02/2016 04:44:51PM     93        5690     systest+3^systest
     %SYS
9     41681038     11/02/2016 04:43:34PM     93        5690     systest+3^systest
     %SYS
9     41681038     11/02/2016 04:42:06PM     93        5690     systest+3^systest
     %SYS
9     41681038     11/02/2016 04:41:21PM     93        5690     systest+3^systest
     %SYS
9     41681038     11/02/2016 04:39:29PM     93        5690     systest+3^systest
     %SYS
9     41681036     11/02/2016 04:38:26PM     93        5690     systest+3^systest
     %SYS
9     41681036     11/02/2016 04:36:57PM     93        5690     systest+3^systest
     %SYS
9     41681036     11/02/2016 04:29:45PM     93        5690     systest+3^systest
     %SYS
```

It may seem obvious, based on the column headings, what each item in an entry is, but I will describe them all.

Printing the last 8 entries out of 8 total occurrences

While this is not part of a syslog table entry, it is an important thing to look at, so I will mention it. This is where you look to find out how many entries have been made to your syslog table. In this example, there have only been 8 entries made since Caché was started. This is my test system, so there are not many entries. If you see ‘ Printing the last 500 entries out of 11,531 total occurrences ’ , you know that you have missed a lot of entries. Increase the table size to the maximum of 10,000 if you are interested in seeing these entries, or run SYSLOG more frequently.

Err

This is the information that we log about the event of interest. People often think that it is always an OS level error from /usr/include/errno.h, (unix), and it often is, but not always. It can really be anything we want to log. For example, it can be debugging information from a diagnostic adhoc, the value of a C variable, or our own error code definitions, (anything greater than 10000). How do you tell what it is? You really have to look at the line of C code that is indicated by **mod** and **line** in the entry. This means that you cannot tell what it really is, without contacting InterSystems. Why bother looking at it then? Well, there are a lot of things that you can figure out without knowing exactly what **err** is, by looking at the other information. You can also contact InterSystems if you notice that there are lots of entries, or entries that are different than you usually see. Keep in mind that an entry in the syslog table is not always an error condition.

Process

This is the process ID of the process that put the entry into the syslog table. For example, if you have a stuck, spinning, or dead process, you can look in the syslog table to see if it logged anything. If it did, it will probably be an important clue as to why the process got into trouble.

Date/Time

This is the date and time that the entry was made. It is very important to correlate the date and time of the entry to the date and time of any system events, because it is often a clue as to what went wrong.

Mod and Line

Mod corresponds to a specific C file, and line is the line number in that file that put the entry into the syslog table. Only InterSystems staff with access to the kernel code can look this up. Only by looking up this code can you tell exactly what is logged in the entry.

Routine

Tag, offset, and routine that the process was running when the entry was made to the syslog table. This can really help to figure out what is going on.

Namespace

This is the namespace that the process was running in.

So, how can I figure out why err 9 is in my syslog table?

First, look at the routine indicated. Here is my ^systest routine:

```
systest  ;test for syslog post
         s file="/home/testfile"
         o file:10
         u file w "hello world"
         c file
         q
```

The syslog entry indicates that systest+3 is what was running when the entry was made. This line is:

```
u file w "hello world"
```

Since the process was trying to write to a file, this might really be an OS level error, so look in /usr/include/errno.h for 9, and find:

```
#define EBADF   9       /* Bad file descriptor                  */
```

Since 9 is about files, and since the line of code indicated is trying to write to a file, it is reasonable to guess that this really is an OS error code.

Can you figure out what is wrong?

To solve this, I first looked at the permissions on the /home directory and on the testfile file. They were both 777 so I really should have been able to open and write to that file. Upon closer look at my code, I noticed an error. The

ten second timeout should have been preceded by two colons, and I also needed to use some parameters in the open command. Below is the updated routine, which actually finishes without errors and writes to the file:

```
systest  ;test for syslog post
         s file="/scratch1/yrockstr/systest/testfile"
         o file:"WNSE":10
         u file w "hello world"
         c file
         q
```

Summary

The syslog table is a valuable tool for debugging if used correctly. Keep the following in mind when using it:

1. **err** is not always an operating system error, it can be anything that we want to log. Contact InterSystems to find out what we logged.
2. Use the other information that is logged to determine what is happening. The line of COS code combined with the error can leave you with a reasonable assumption as to if it is an OS error.
3. Look in the syslog table any time you are having a problem you cannot solve, as there may be clues there.
4. Use the **Date/Time**, number of entries, and total occurrences to determine if you need to increase the size of your syslog table.
5. Be familiar with what your system is logging to the syslog table, so that you know if changes or new/different entries occur.
6. Entries in the syslog table are not necessarily a problem.

#Best Practices #Caché #Terminal #InterSystems IRIS #Monitoring #Tips & Tricks