
Subject: Re: Sysvars

Posted by [Ken Mankoff](#) on Tue, 16 Aug 2005 23:42:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Tue, 16 Aug 2005, Reimar Bauer wrote:

> do we have a routine which is able to store all available sysvars
> in a script to a structure (without the need of execute)?
>
> help,result.!x,/str

I do not know of any. I just spent a short while looking into this and it seems like a tricky problem. I'd be interested in your solution when (if?) you get one.

You can get a list of all sysvars like this:

help, /sys, out=out

You can get the names and types of the fields of a sysvar (or any structure) like this:

CALL_PROCEDURE, 'help', names="!CPU", /st, out=fields
and if nothing is returned then you know that the var isn't a structure (like !PATH or !C)

At this point you can re-create each as a variable/structure that can be passed into the CREATE_STRUCT used to combine all your structures. I have code that creates a structure called "all" that contains substructures of every system variable. This is almost what you requested.

What I cannot do is populate the values of the re-created structure.

CALL_PROCEDURE, 'help', names='!CPU.HW_NCPU'

returns nothing. It isn't just the structure that causes this problem.

CALL_PROCEDURE, 'help', names='!PATH'

is also silent.

Likewise, SCOPE_VARFETCH('!CPU') won't work to get the values contained by this structure.

-k.
