

Programming UMDs Using Auxiliary Variables

Auxiliary variables are global variables used to program control strings for UMDs and GPIs (collectively called “status devices”) outside the normal status device edit fields. The advantage of using auxiliary variables is that the main UMD or GPI contents field is left free for the user to enter his or her own information without fear of disturbing the UMD or GPI control string. Information entered into the contents field of a status device can be read back from within an auxiliary variable control string and used as required.

Status devices for which no auxiliary variable is programmed are programmed in the normal way, by entry of text and embedded functions into the contents field.

Auxiliary variables are associated to a particular status device by the name of the variable.

Auxiliary variable names for UMDs have two possible formats:

@SDV<Two-digit interface number><1-digit Section Number><Serial Number>

Or

@SDV<Two-digit interface number><1-digit Section Number><UMD name>

The two-digit interface number is valid in the range of 1 to 56. The one-digit section number is valid in the range 1-3. The serial number is actually a name which must match the text of the UMD serial number field in the TSC.

Example 1:

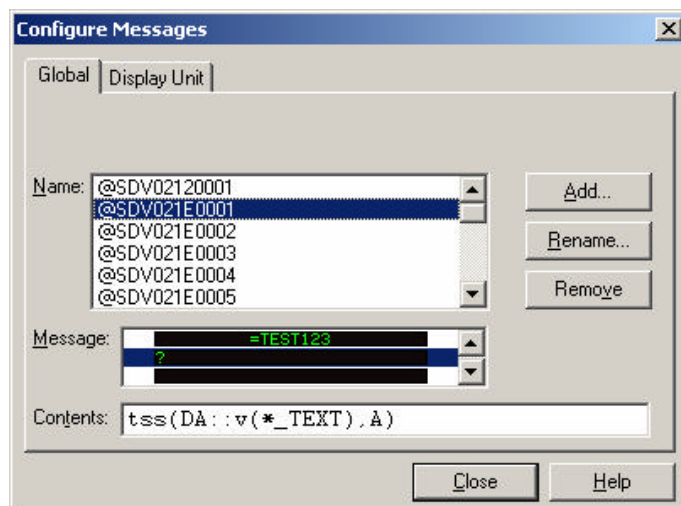
@SDV02155555

is the name of an auxiliary variable associated with a UMD section 1 on interface 2, with serial number 55555.

Example 2:

@SDV021MON051

Here the name of the UMD rather than its serial number is used. This is particularly useful for Image Video UMDs, allowing UMDs of different serial numbers to be swapped out without renaming the associated auxiliary variable.



Auxiliary variables for GPI outputs have the format:

@SDV<Two-digit interface number><2-digit Port Number><4-digit GPI Output Address>

Again, the two-digit interface number is valid in the range of 1 to 56. The port number is valid in range 1-12. The GPI output address is valid in the range 1-512.

Example:

@SDV02070001

is the name of an auxiliary variable associated with a GPI output on port COM7 at GPI output address 1.

The first thing that one notices after having programmed an auxiliary variable for a UMD is that typing text into the UMD contents field has no effect and the UMD display remains blank. The contents field for every UMD or GPI output is read into a static variable named “*_TEXT”. To make the UMD (or GPI output) reflect what is typed into the contents field simply program the auxiliary variable for the status device with “v(*_TEXT)”.



A more useful example of the use of auxiliary variables would be to set the auxiliary variable of a UMD with something like “TSS(DA::v(*_TEXT),A). v(*_TEXT) will return whatever has been typed into the UMD contents field. With this programming the user can quickly reprogram the source tallied by a UMD by simply typing the name of the source directly into the UMD contents field (for example “CAM01”). There is no danger of disrupting control strings with an ill-placed or missing parenthesis. The same principle applies to GPI outputs.