



# Driver Variables

The following variables are used or set by the driver programs and may be set or checked by various procedures as indicated below.

**CREATING** – This variable is set by the file maintenance driver and signals that a new record is being created. This variable differs from the usage of the variable Z[6]. Z[6] is set to zero while the user steps through the fields the first time, but upon re-entering a field, Z[6] is set to 1 or 2.

**GO\_BACK** – This variable is initialized to -1 in the entry drivers (SMC901 and SMC902) prior to the validation procedure being executed. The validation procedure should set the variable to 0 if the value in the field is valid or to 1 if the value is invalid.

**GO\_BACK** should not be changed in an F1 – F4 procedure unless focus must be returned to the field after the procedure is finished. In that case it should be set to 1.

In the following example, **GO\_BACK** will be set to 1 if X\$ does not contain a valid AP GL Table.

```
1000 VAL_AP_GL_TABLE: ! 1000
1005 setesc 9710; seterr 9810
1010 if SMCNTL=0 then gosub OPEN_SMCNTL
1020 let GL_TABLE_DESC$="Not on File"; find (SMCNTL,key="APG"+%A0$+X$+"0",
1020:dom=*next)*,*,*,GL_TABLE_DESC$; let GO_BACK=0; goto *return
1030 let GO_BACK=1
1040 return
```

**SKIP\_INPUT** – This variable is checked by the entry drivers and should be set in a pre-input procedure. Setting it to 1 tells the driver that the input field is to be skipped. This is primarily of use in the CUI interface. Causing a field to be skipped under certain circumstances is most effectively accomplished through setting the Edit Field Eval or the Disable Condition in the SMPRMT record.

**FAILED** – This variable is checked by the file maintenance driver immediately following performing the initialization procedure. The

initialization procedure should set FAILED=1 if the file maintenance should not be run, but instead return to the menu.

In the example below, the file maintenance will cease to continue if the "SO to GL Posting Table" is not set up.

```
20000 INIT_PROG: ! 20000
20010 let SMCNTL=hfn; open (SMCNTL)"data/SM/SMCNTL"
20020 let SOSVIA=hfn; open (SOSVIA)"data/SO/SOSVIA"
20030 let FRT_GL_CODE$="";find(SMCNTL,key="SOG"+%A0$+"0000",
20030:dom=*next)*,*,*,*,*,FRT_GL_CODE$; goto *return
20040 let MESSAGE$="SO_TO_GL",FAILED=1
20090 return
```

**REDISPLAY** – This may be set in a validation routine, and it indicates to the driver programs that they should initiate the process of redisplaying the SMPRMT entries that are marked "Update Field".

**MESSAGE\$** - This variable may be set at various points, including validation procedures, initialization procedures, etc., to have the driver issue the message indicated. The variable should be set to the message code from either SMMSG\$ or SMCMSG\$, but it may be set to a literal message.

**USE\_PICKLIST** – When setting up an SMPRMT record which has Z4\$ Eval set to a set of valid values, the GUI interface will create a Drop Box. You can set PICKS\$ to a list of descriptions which will appear in the GUI drop box. In order to have those descriptions also appear in a pick list on the CUI interface, you must set USE\_PICKLIST=1 in the pre-input procedure.

**PICKS\$** - Set this in a pre-input procedure to assign descriptions to the various options in a Drop Box or Pick List.

**FROM\_KEY** – This is set and used by the driver programs to determine that they are interacting with the key elements of a file.

**BEGINNING\_TAB** – This should be set prior to calling SMC901 or SMC902. It is the first Tab Order that the entry driver should access.

**ENDING\_TAB** – This should be set prior to calling SMC901 or SMC902. It is the last Tab Order that the entry driver should access. When wrapping from the beginning to the end of the screen, the driver uses these variable to determine what the beginning and ending is.

**START\_TAB** – Also set prior to calling SMC901 or SMC902, this is the initial Tab Order which should receive focus.

**TAB\_SCREEN** – This is set by the drivers and should never be changed by a non-driver program. It is the Tab Number that is currently being accessed.

**GOT\_CTL** – This is the CTL value as set/returned by the entry drivers (SMC901 and SMC902). It can be manipulated in function key procedures or validation procedures to force the drivers to change their actions.

**KEY\_ELEMENTS** – This is set by the drivers, and indicates the number of key elements present. It is primarily used in the file maintenance drivers. The first tab order accessed after the key elements have been entered is always 10+KEY\_ELEMENTS.

**FILE\_NAME** – This is the channel on which the primary file in the file maintenance driver is opened. You should never use or change this variable.

**CUST\_FILE** – This is the channel on which the companion file in the file maintenance driver is opened. You should never use or change this variable.

**F\$** and **ORG\_F\$** - In the file maintenance driver, F\$ represents the current state of the primary file record. ORG\_F\$ is the original copy of the record. Comparing F\$ to ORG\_F\$ tells the driver whether the record has changed.

**CF\$** and **ORG\_CF\$** - Also in the file maintenance driver, CF\$ represents the current state of the companion file record. ORG\_CF\$ is the original copy of the record. These are also used when determining whether the record has changed.

**SMPRMT** vs. **%SMPRMT** – The drivers use SMPRMT as the channel on which the SMPRMT file is opened. If SMPRMT is zero, it will use the global channel %SMPRMT. This allows the drivers to open an alternate copy of SMPRMT (as in the 3-Level Entry Driver) and have the drivers use it instead of the primary SMPRMT file.

**TABS** – This is set and maintained by the drivers. It is the number of Tabs defined.

**LINE\_TO\_EDIT** – This is set prior to calling SMC901 from the CUI file maintenance driver to tell the entry driver that the user is only editing an individual line.

**\_MENU\_BAR\_MAIN\$** - This is set to the syntax of the ProvideX menu\_bar command for the main options of the menu bar.

**\_MENU\_CTL** – This is the CTL value that is assigned to the menu\_bar. When the entry drivers see this ctl value, it returns to the calling program so it can deal with reacting to the menu selection.

**\_MENU\_BAR\_SUB\$** - This completes the syntax started in \_MENU\_BAR\_MAIN\$, and includes the portion of the menu\_bar command that creates the sub options of the menu selections.

**\_MENU\_BAR\_STATUS\$** - This variable is used to track which menu selections are active vs. inactive and checked vs. unchecked.

**DEFAULTS** – This is used by the CUI entry driver to “fast forward” through the prompts. It can be set in a validation routine of a function routine. The driver will “fast forward” until it completes all tab orders between BEGINNING\_TAB and ENDING\_TAB or until it encounters a GO\_BACK condition.

**BACKUPTO** – In the CUI interface, you can set BACKUPTO to the Z[9] value of the field you want to go back to, then set Z[11]=4. This is done in a validation routine of a function key procedure.

**COMPRESSED** – This variable must be set to 1 in the compress procedure to indicate to the driver that the compress procedure actually changed the value in X\$. If it did not change X\$, it should be set to 0.

**EXPANDED** – This variable must be set to 1 in the expand procedure to indicate to the driver that the expand procedure actually changed the value in X\$. If it did not change X\$, it should be set to 0.

**TABNUM** – This variable is maintained and used by the entry drivers to maintain which Tab Order currently has focus. This variable should never be changed.

**END\_VALIDATION** – In the GUI interface, when the user clicks the OK button SMC902 automatically performs the validation routines for all of the fields on all of the tabs. When that happened, END\_VALIDATION will be set to 1 so the validation routine can act differently if necessary.

**\_F4\_OK** – Used by the GUI entry driver to indicate that the F4 key should not wrap around to the end but should exit driver.

**NEXT\_BTN** – This is the CTL value of the button that selects the next record in the file maintenance. When SMC901 or SMC902 receive this value, it automatically returns to the calling program with GOT\_CTL = NEXT\_BTN.

**PREV\_BTN** – This is the CTL value of the button that selects the previous record in the file maintenance. When SMC901 or SMC902 receives this value, it automatically returns to the calling program with GOT\_CTL = PREV\_BTN.

**SAVE\_BTN** – This is the CTL value of the button that is either Ok or Save. When the drivers get this ctl, it automatically performs the validation routines for all the fields and exits to the calling program.

**\_RETURN\_CTL\$** - This is a list of other ctl values that should cause the entry driver to return to the calling program. They should be set like this: `_return_ctl$=str(-1019:"-####")+str(300:"-####")`

**EPROG\$** - This variable should be set to the program name from SMPRMT that the entry drivers or the Answer/Entry Driver should interact with. Set it prior to calling them.

**PCONDS\$** - This variable should be set to the runtime condition from SMPRMT that the entry drivers or the Answer/Entry Driver should interact with. Set it prior to calling them.

**\_AUTO\_EXIT** – This variable can be set prior to calling the entry drivers or the Answer/Entry Driver to indicate that the entry driver should automatically exit to the calling program when the ending tab order has been reached, instead of wrapping back to the beginning of the tab order.