

Event Subroutines

Version 1.0 31Mar06

The traditional way to customize a TPH program is thru what I named “Code Segments” – these are the snippets of code that are used as a template when writing the program. You can modify these for a particular program to add special processing, validations, etc.

Other tools use a different techniques and one come one is the ability to call a user written subroutine at key points in the process. This technique is common with those tools that don’t actually write code but are driven by a central run time.

Due to popular request, I have added some of that functionality to TPH. I call the subroutines “Event Subroutines”

Currently you have to write these by hand, but I will be adding a “wizard” to help with this.

For each data entry field there are three places that a subroutine can be called: When the field is “activated”, to perform a validation, and when the field is “deactivated.”

The validation subroutine is handled a little differently than the other two. To use a validation subroutine you choose a FIELD TYPE of “SU” (Subroutine) and place the name of the subroutine in the VALIDATION field.

The activate and deactivate subroutines are entered via the fields ACTIVATE SUB and DEACTIVATE SUB found on the “More” tab at the bottom of the field detail screen.

In addition you can define a subroutine call in the “Update Program Definition” screen under the tab “Event Routines.” These are called “Program Events” as opposed to the “Field Events” described above. There are eight events you can trap:

- Initialization (at the start of the program)
- Pre-read (before the record is read)
- Post-read (after the record is read)
- Pre-write (before the record is written)
- Post-write (after the record is written)
- Pre-delete (before the record is deleted)
- Post-delete (after the record is deleted)
- Close (just before the program ends)

Structure of the Event Subroutine

Since you only specify the subroutine name the parameters are fixed a typical subroutine might start out:

```
SUBROUTINE TPH.CHECK.VALIDATION(EVENT, Value, item.id, MAT
STEPS.REC, MAT WORK.REC, WorkList, options, ReturnFlag, ReturnString)
```

The meaning of the parameters may change somewhat with the type of event. This will be described in more detail later.

Notice that the first parameter is the EVENT type. This means you may write one subroutine to handle all of the events for a particular field. You can of course, write individual subroutines as well.

<u>Parameter</u>	<u>Description</u>
Event	Event triggering the call. Can be one of; ACTIVATE, VALIDATE, DEACTIVATE, INITIALIZE, PREREAD, POSTREAD, PREWRITE, POSTWRITE, PREDELETE, POSTDELETE, CLOSE
Value	For field events this is the current value of the field. It is null for program events.
item.id	The Item ID of the current record. It is null for the INITIALIZE and CLOSE events.
MAT STEPS.REC	This is the record of the main file. It will be all null for the INITIALIZE and CLOSE events and for any field events that are part of the Item ID. The name of the array will, of course, be different for different files.
MAT WORK.REC	Contains the values of any Item ID fields, any WORK fields you've defined in the program. The actual order of the work fields is indeterminate so you need to use the next parameter to find a particular field.
WorkList	A multivalued list of work field names. WorkList<1,1> corresponds to WORK.REC(1)
options	A user defined field you can use to pass additional information to the event subroutine.
ReturnFlag	A Boolean (True/False) that is returned by the event subroutine. Its actual meaning depends on the event being processed.

ReturnString	A string that may be displayed as an error or informational message. Its mean depends of the event being processed.
ReturnFlag/ReturnString meanings (if an event is not listed then ReturnFlag has no effect):	
ACTIVATE	If ReturnFlag is set to FALSE then ReturnString will be displayed (if it has a non-null value) and the user will not be allowed to change the field.
VALIDATE	If ReturnFlag is set to FALSE then ReturnString will be displayed and the current field will be re-activated.
DEACTIVATE	If ReturnFlag is set to TRUE it means the deactivate event subroutine changed data and the current screen/form should be redisplayed.
POSTREAD	TPH programs set a variable new to TRUE if this is a record is not found on file. The POSTREAD event subroutine is passed this value and may change it.
PREWRITE	If ReturnFlag is set to FALSE the write will be abandoned and the ReturnString displayed (if non-null)
PREDELETE	If ReturnFlag is set to FALSE the delete will be abandoned and the ReturnString displayed (if non-null)