

## Random notes on the new TPH

The number of parameters need for each step has gotten so great that the concept of the screen def as a text item doesn't work anymore. There is a new dedicated screen painter that works off of what was the "compiled" screen item (e.g. %UPD.CUST in DICT CUSTOMERS). Some things now take a little more time, others are faster. All in all it's an improvement, I think.

When you define a program like UPD.CUST when you are all done you will have:

UPD.CUST.GS	("Green Screen" – i.e. the old terminal based pgm)
UPD.CUST.ATG	("AccuTerm GUI – the new gui version)
UPD.CUST.CALLER	(the calling program – will call .ATG on an accuterm terminal)

In the future there will also be UPD.CUST.WEB (and who knows what else ☺ )

When doing a file validation, no "return value" (e.g. Customer Name) is displayed. You must create a new "step" with a validation type of D ("Display field"). The upside is you can now return many fields from one read, each as its own step. This is the one area where existing screens defs will need to be modified.

The "Input" program segments (UPD.SIMPLE, UPD.FILE, etc) have been split a bit, in order to make code re-usable between GS at ATG programs. There are now four pieces to one input...

ACTIVATE	(before any input. Saves orig value, etc.)
INPUT	(the call to TPH.INPUT and the LOOP UNTIL CORRECT logic. Does not exist for ATG – that is done by the GUI driver)
VALIDATE	(Checks if on file, etc.)
DEACTIVATE	(Does any calculations, displays any "display" fields, etc.)

Sequence of operation:

- Define the program (as before)
- Create the GS layout (with new painter)
- Generate, compile and test GS program
- Build GUI layout from GS. This creates an item that is compatible with the AccuTerm GED editor, which can be used to change look of the GUI screen (move things around, change fonts, colors etc.)
- Generate, compile and test ATG program

Right now I have no easy way to modify the TPH parameters of the GUI screen. I had thought to use an option Pete put in the GED editor to write my own property pages, but right now he doesn't expose all the fields I need. So the sequence is to change the GS definition and then "update" the ATG with the changes. This may not be a bad thing, as it

always keeps the two in synch. When you do this, you won't lose the cosmetic changes you've made with GED editor.

There are now many different levels of "Code Segments" (i.e. modifications of the default program):

File Level:

<filename>\*POSTREAD      will be inserted whenever the file is read, even if it is just being read to display fields.

<filename>\*PREWRITE      will be inserted just before the write (to reverse anything done in POSTREAD)

<filename>\*POSTWRITE      will be inserted after the write.

Field Level:

<filename>\*<attr name>\*VALIDATE  
Will be used in *any* program that uses this attribute. Similarly for ACTIVATE and DEACTIVATE

Program Level:

<program>\*<attr name>\*INPUT      This is the "INPUT" segment that is only used for GS programs

<program>\*<attr name>\*VALIDATE  
A custom validate routine for this program only. Similar for ACTIVATE and DEACTIVATE.

Note: when I say "program" I mean the base program, e.g. "UPD.CUST" -- there is a code sgmt variable that you can use to include code only for GS or ATG.

The idea is to be able to re-use code customizations between the different platforms. Obviously this can't be 100% but it should be pretty close.

More to come....

***Added 11/22/04***

Using the new screen painter.

The main difference is the only painter just updated a text item with no logic built in. The new one updates the "compiled screen" item directly.

The cursor movement keys work the same. D deletes a char, I inserts a space (different than before) and T still lets you enter text (now called “free text” – not tied to a data entry field)

New stuff:

A fwd slash (/) is to update a data entry field. If over an existing one it will take you to the step window for that field. If over a “blank” it will create a new field with the prompt position at where the slash was entered.

A back slash (\) will delete the data entry field or free text. You are asked to confirm.

A semi-colon (;) will let you move a field. You move the cursor to where you want (using the arrow keys) and hit ; again. The input entry position will move to that point. I haven't really figured out how to handle the Prompt Test – the logic now is kind of strange and it may seem to move to arbitrary locations but there is logic ☺ Best bet is to edit the field (/) and change the prompt line and pos

Some non-obvious prompts on the steps window:

File Name – for all by validation type D (Display field) this will show the main file name. For validation type D it will let you enter the file name to retrieve from.

Read Item ID. Also just for display fields. Which field is used to read File Name (for example, you may have a “bill to” and “ship to” customer in one screen but both are validated against the customer file.

Validation. If T (Table Validation) this is the table name. If F (file) this is a translate (probably should be the file name but haven't gotten around to changing it). If C (Conv) this is the conversion. If S (single character) this is the list of characters (i.e. YN, ABC, etc.). If N this is the named pattern.... And so on.

MV Type: This is something like V10, S5 etc – the field after the 2<sup>nd</sup> colon on the old screen item.

Step# : This is the step number for the field. The pgm will try and renumber going top to bottom and then left to right. It keeps the same order though. That is, if you add a 2<sup>nd</sup> step 3, the old step 3 becomes 4, the old 4 becomes 5 etc down the line. Can be “4.1” “4.2” etc for MV fields.

There is a lot of work I need to do here for validation and error checking, but that will have to wait a bit.