

User Interface - Bug #2528

finish investigation and implementation of INPUT, SCREEN-VALUE and INPUT-VALUE

03/02/2015 11:22 AM - Constantin Asofiei

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No	version:	
vendor_id:	GCD		
Description			
Related issues:			
Related to Base Language - Bug #2188: fix runtime for SCREEN-VALUE attribute		New	
Related to User Interface - Bug #2767: implement INPUT-VALUE attribute proper...		Rejected	

History

#1 - 03/02/2015 11:30 AM - Constantin Asofiei

This task is meant to complete the implementation of the INPUT, SCREEN-VALUE and INPUT-VALUE.

Some initial findings which can be used as an investigation start:

1. The INPUT function looks like it acts (to some extent) as a `_POLY`:
 - can be used as a CHARACTER (r-value in an assignment to a char var, parameter, etc)
 - otherwise, the type of the expression in which is used (parameter, r-value, etc) must match the widget's data type.Determine how this behaves in these contexts: is it the same as SCREEN-VALUE when used as a character? Internally the conversion from CHARACTER to `<expected-type>` takes place and is enforced (via compile-time error) only when the INPUT function is used in a non-character expression (similar to assigning a `_POLY` result to a var). In P2J this is not the case: INPUT function is always strong-typed.
2. more, for a DATE variable, it's SCREEN-VALUE/INPUT function (when used as a character) can return an invalid value, like "11 / / ", after F4 is pressed and condition is raised. Validation of the INPUT value is not done until it needs to be cast to date.
3. check how format affects them; does INPUT function return the formatted value when assigned to a character?

#2 - 03/23/2016 04:10 PM - Greg Shah

- Target version deleted (Milestone 12)

#3 - 03/23/2016 04:10 PM - Greg Shah

- Project changed from Base Language to User Interface

#4 - 07/18/2016 02:20 PM - Greg Shah

A simple version of INPUT-VALUE has been implemented. It has been mapped to the getter for each widget, which is the same implementation as INPUT. Both are POLY implementations. In both case we have some open questions about whether we are 100% compatible.

Some time ago (June 2008), we did an investigation of INPUT, SCREEN-VALUE, ASSIGN and INPUT-VALUE. The test programs are not well automated and I think some results (related to format strings) were mis-interpreted because of the use of frames and message statements to display results of reading screen buffer values. The testcases can be found in `testcases/uast/input_value_adjustment*.p` and the findings from that time are in `testcases/uast/input_value_adjustment.html`. We should automate these cases and rework things to make them unambiguous. One example of

how to automate is testcases/uast/data_types_created_from_question_mark_string.p, but the better example is in testcases/uast/library_calls/*.

Some things that need to be explored for INPUT-VALUE:

- the full range of widgets
 - widgets that are expected to store data: fill-in, editor, combo-box, browse (cell), radio-set, sel-list, slider, toggle-box, text, literal
 - widgets that are not supposed to work (e.g. buttons...)
- all BDT data types that can be mapped to a widget
- unknown value
- at-base fields (what input value do you get back when the screen-buffer was assigned from a different data type using the at-base field syntax?)
- uninitialized widgets
- automatic conversion of POLY return values (e.g. dynamic-function)

#5 - 10/07/2016 12:37 PM - Constantin Asofiei

Using uast/screen_value/screen-value-overrides2.p, see how the toggle-box reacts when SCREEN-VALUE is set to either empty string or unknown. The problem I think is in GenericFrame.setScreenValue, which converts from character to logical, and passes the incorrect logical value down to the client-side.

#6 - 10/07/2016 12:39 PM - Constantin Asofiei

Another issue found in screen-value-overrides2.p is that the behaviour is different between GUI and ChUI, for some widgets. Also, BROWSE:SCREEN-VALUE needs to be tested in a similar way as the other widgets.