User Interface - Feature #12053

replacements for the smaller DBI Component Toolbox controls

10/20/2026 06:04 PM - Greg Shah

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**reviewer:**	
vendor_id:	GCD	**production:**	No
case_num:		**env_name:**	
version_reported:		**topics:**	
version_resolved:			

Description

History

#1 - 10/20/2026 06:04 PM - Greg Shah

Replacements for the smaller DBI Component Toolbox controls

Project: User Interface

Problem

Several DBI Component Toolbox controls map onto an existing FWD or 4GL widget with little more than an alias table:

Control	Used for	Members and events	Target
ctMeter	a property-driven gauge or progress meter	VALUE, ForeColor, VISIBLE	PROGRESS-BAR
ctImage	a passive icon store feeding menus and lists	ListImage(n), AddImage, RemoveImage, ImageListCount	IMAGELIST
ctDate	a month calendar	DATE, Weekends, DateBackColor, DateChange	CALENDAR
ctCalendar, ctMonth	a month view with per-day text and images	per-day text and images	CALENDAR
ctCombo	a combo box	TEXT, AddItem, KeyPress	native COMBO-BOX
ctSlide	a slider for a numeric value	VALUE	native SLIDER
ctTabs	a tab strip	Caption, SelectedTab, Click	TABSET
ctListBar	a graphical list bar used as a filter selection	AddList, AddImageItem, SelectedItem, ItemClick	native SELECTION-LIST, or the ctList list grid

Work

1. Map each control onto its target, with the alias and event tables, and the design-time state from the .wrx (for example the images embedded in a ctImage).
2. Extend the targets where a member above has no counterpart, such as per-day text on CALENDAR.
3. Register the control's CLSID in the OCX registry (ocx-registry.xml), mapping it to the target widget, as #11919 did for ctToolBar.
4. Decode the design-time state the control keeps in the .wrx storage at conversion time, so that it exists after conversion.
5. Implement the members and events listed above on the target, with the vendor's semantics: argument types, return values, defaults, and the order in which events fire. Members of the type library that are not used become no-ops with a conversion warning, as the OCX pipeline already does.
6. Follow the OCX emulation playbook attached to #11919: an OpenEdge test program with goldens for each member and event, recorded input sequences for the interactive parts, and Lyra automation.
7. Update the gap marking and the OCX registry documentation.

Testcases

Each member and event above, exercised on Windows OpenEdge with the real control to produce goldens, then run against the FWD replacement on the web and Swing clients. The vendor help files are the reference where behaviour cannot be observed directly.