User Interface - Feature #12046

replacement for the DBI ctDropMenu ActiveX control (context menu)

10/20/2026 05:48 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 05:48 PM - Greg Shah

Replacement for the DBI ctDropMenu ActiveX control (context menu)

Project: User Interface

Problem

ctDropMenu (ctDropMenu.ocx 1.0) is a right-click context menu that applications rebuild on every use: ClearItems, AddItem, ItemCtrlID, ItemPicture, ItemEnabled and ItemSubText, then DropMenuAtCursor, which **blocks** until the user picks an item. The ItemClick handler stores the chosen id in a variable, which the calling code reads after DropMenuAtCursor returns. The .wrx holds no menu content.

FWD supports dynamic 4GL popup menus, but they are event driven and do not block.

Work

1. Implement an emulation class exposing the ctDropMenu API over FWD's dynamic popup menus.
2. Make DropMenuAtCursor block with a nested event loop, as an alert box does, so that ItemClick runs before it returns, and return when the menu is dismissed without a choice.
3. Support item images (typically from a ctImage or an image list) and disabled items.
4. Register the control's CLSID in the OCX registry (ocx-registry.xml), mapping it to the target widget, as #11919 did for ctToolBar.
5. Decode the design-time state the control keeps in the .wrx storage at conversion time, so that it exists after conversion.
6. 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.
7. 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.
8. 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.