

User Tips & Techniques

Using UDCs to Control Program Behavior

By Jeffrey S. Geiger, Oracular

OFTEN, IT BECOMES NECESSARY TO MAKE PARTICULAR VERSIONS OF J.D. EDWARDS PROGRAMS BEHAVE DIFFERENT from one another. One option is to add additional processing option values to the Processing Option template and pass in the values that control how it works. Sometimes that is not possible or not logical. This article will present an alternative method for accomplishing this goal in an extremely easy fashion.

Solution

The problem with modifying processing options can be that the processing options are referred to in other programs, they could be fetched by a business function in another program (this causes hard errors) or you just don't want to modify another standard J.D. Edwards object.

If you really need one version to behave different from another and adding fields to the existing processing option template just won't do, there is another direction to go. All you need to do is create a custom UDC with the version name as the UDC key (KY). Let's say you want to make the P4210 — Sales Order Entry show a custom field on it only for certain versions and hide it for others.

All you would need to do is the following:

- Create a new UDC — for example — 55IPP
- Make it a code length of 10 (this will hold the version name)
- Check the "2nd Line" check box.

Enter rows into the UDC table where the Code (KY) is equal to the versions of the program you are running. See below for an example.

Within the event rules of the program, you will perform the following:

- Get UDC to fetch the Description 2
- Use the Description 2 to determine how to proceed

Example:

Using the value SL Version Name (this is a system variable), get the UDC by calling business function GetUDC – X0005. Map the following parameters (note direction):

```
55 → szSystemCode
PP → szRecordTypeCode
SL Version Name → szUserDefinedCode
VA rpt_ShowHide ← szDescription002
```

With the value fetched from the UDC, you can now check if you should do what you want to do.

```
If VA rpt_ShowHide = "Show"
    Show the control
Else
    Hide the control
End If
```

Frankly, you can put whatever value in the UDC you want to and do whatever you want to within the event rules. It is entirely up to the person writing the event what behaviors you want the program to exhibit.

Conclusion

For many developers the first inclination is to modify the processing options whenever you want a certain version of a program to behave differently from another. When you want to avoid modifying another object (Processing Option Data Structure) or are restricted from doing it for one of several reasons, making a UDC that turns flags on and off within the program is a simple way to add flexibility to your programs. □

Jeffrey Geiger is president, Oracular Michigan, Inc. and can be reached at jgeiger@oracular.com. For more information about Oracular, visit www.oracular.com.

CODES	DESCRIPTION 1	DESCRIPTION 2	SPECIAL HANDLING	HARD CODED
VER0001	Special Version of P4210 1	Show		N
VER0002	Special Version of P4210 2	Hide		N
VER0003	Special Version of P4210 3			N