

#16: CUST Resource

Written by: Samir Arora
Apple Computer, Inc.

October 6, 1987

This note describes the CUST resource in 4D.

There is a resource of type "CUST" with ID number = 0 in 4th Dimension that contains a number of parameters that 4D uses. These parameters can be modified using ResEdit, or External Procedures.

The structure of CUST is:

CUST = Record

```
DAsInPrinting:    INTEGER;
DAsInThermometers: INTEGER;
CursorSpinRate:   INTEGER;
NewSubRecASCII:   INTEGER;
NewSubRecModifiers: INTEGER;
EnterRecASCII:    INTEGER;
EnterRecModifiers: INTEGER;
CancelRecASCII:   INTEGER;
CancelRecModifiers: INTEGER;
ValidateDialogASCII: INTEGER;
ValidateDialogModifiers: INTEGER;
CancelDialogASCII: INTEGER;
CancelDialogModifiers: INTEGER;
WindowPosition:  INTEGER;
TopWindow:       INTEGER;
LeftWindow:      INTEGER;
BottomWindow:    INTEGER;
RightWindow:     INTEGER;
ResizeWindInEnviron: INTEGER;
ResizeWindAfterMenu: INTEGER;
Reserved1:       INTEGER;
Reserved2:       INTEGER;
Reserved3:       INTEGER;
Reserved4:       INTEGER;
Reserved5:       INTEGER;
CancelDuring:    INTEGER;
Reserved:        INTEGER;
DiacriticalSensitive: INTEGER;
```

end;

The value of Modifiers used in CUST can be the following:

```
NoModifier = 0;
CommandKey = 256;
```



ShiftKey = 512;
AlphaLock = 1024; { Caps Lock }



1. DAsInPrinting: 0 or 1. Default 0: DA's are not accessible during printing. Set to 1 for enabling using DA's during printing.

For running 4D in the background in MultiFinder, this must be set to 1.

2. DAsInThermometers: 0 or 1. Default 0: DA's are not accessible during thermometers, i.e. the window during a Search. Set to 1 for enabling using DA's during thermometers.

For running 4D in the background in MultiFinder, this must be set to 1.

3. CursorSpinRate: The spin rate of the Beach-Ball cursor during procedure execution. If it is 0 then the Beach-Ball does not appear. A positive or negative not equal to 0 specifies the spin rate, with the rate being modulo 32 of this value.

Normal value to set is 16, Default value is 0.

4. NewSubRecASCII: The ASCII code of the character to enter a new subrecord.

Default value is 9 (Tab) with modifiers set to the Command key.

5. NewSubRecModifiers: The value of the Modifiers related to **4. NewSubRecASCII**.

Default value is 256. (Command key)

6. EnterRecASCII: The ASCII code of the character to enter a record.

Default value is 3 (Enter) with no modifiers.

7. EnterRecModifiers: The value of the Modifiers related to **6. EnterRecASCII**.

Default value is 0. (No Modifiers)

8. CancelRecASCII: The ASCII code of the character to not validate a record.

Default value is 46 (Period) with modifiers set to 256 (Command)

9. CancelRecModifiers: The value of the Modifiers related to **8. CancelRecASCII**.

Default value is 256. (Command key)

10. ValidateDialogASCII: The ASCII code of the character to validate a DIALOG.

Default value is 3 (Enter) with no modifiers.

11. ValidateDialogModifiers: The value of the Modifiers related to **10. ValidateDialogASCII**.

Default value is 0. (No Modifiers)



12. CancelDialogASCII: The ASCII code of the character to not validate a DIALOG.

Default value is 46 (Period) with modifiers set to 256 (Command)

13. CancelDialogModifiers: The value of the Modifiers related to **12. CancelDialogASCII**.



14. WindowPosition: Specifies the default position of the 4D window.

if = 0 then the window size = screen size, and the window title is hidden behind the Menu Bar.

if = 1 (Default) the window size = screen size.

if = 2 then the window is displayed according to the values in the items 15,16,17 and 18.

if = 3 then the window is displayed according to the values in the items 15,16,17 and 18, and is centered on the screen.

15. TopWindow: The value of the Top side of the window.

16. LeftWindow: The value of the Left side of the window.

17. BottomWindow: The value of the Bottom side of the window.

18. RightWindow: The value of the Right side of the window.

{ 15,16,17 and 18 have default values = -1 and can be used only if integer number 14. contains 2 or 3.

19. ResizeWindInEnviron: Allows resizing of the window when changing from Design or User to Custom Environment.

If = 0 (Default) then there is no change.

If = 1 then the window is resized according to the value specified in integer number 14.

20. ResizeWindAfterMenu: Allows resizing of the window after the execution of a 4D procedure called from a Menu.

If = 0 (Default) then the window is not resized.

If = 1 then the window is resized according to the value specified in integer number 14.

21. Reserved1: Reserved by program, do not use.

22. Reserved2: Reserved by program, do not use.

23. Reserved3: Reserved by program, do not use.

24. Reserved4: Reserved by program, do not use.

25. Reserved5: Reserved by program, do not use.

26. CancelDuring: Specifies if the "During" phase is called when the user does not validate the data entered, i.e. if the user clicks a Don't Accept button or types the associated Don't Accept Key with Modifiers.

If = 0 (Default) then "During" is called.



If = 1 then "During is not called.

27. Reserved6: Reserved by program, do not use.

28. DiacriticalSensitive: Specifies if the Search is diacritical sensitive.



If = 0 then the searching is not diacritical sensitive.

Ex. Search([File1]Name="A") will return all records with Name equal to "A", "a", "Å", "å", etc.

If = 128 then searching is diacritical sensitive.

Ex. Search([File1]Name="A") will return all records with Name equal to "A" and "a" only.

There is a resource template for CUST that you can copy into ResEdit, that allows you to open CUST in ResEdit. To install the template do the following, see the Note on installing TMPLs in ResEdit.

Once you have the TMPLs installed in ResEdit, do the following

1. Launch ResEdit.

Now , in ResEdit, you can Open the 4D program, and open CUST resource with ID = 0 and modify it.

The second way to modify CUST is to use the external procedures GetIntegerRes and SetIntegerRes in a 4D database. To use these, see the note on the external procedures for resource manipulation.

The procedures have the following format:

```
GetIntegerRes ( <-ResType:      STRING[4];
                <-ResID:        INTEGER;
                <-WhichInteger:INTEGER;
                ->Value:        INTEGER )
```

```
SetIntegerRes ( <-ResType:      STRING[4];
                <-ResID:        INTEGER;
                <-WhichInteger:INTEGER;
                <-Value:        INTEGER )
```

GetIntegerRes returns the value of the integer in the WhichInteger position, in the resource of type ResType with ID = ResID.

For example:

GetIntegerRes("CUST";0;1;vDAsPrint) will return the value of the 1st integer in vDAsPrint from the CUST resource with ID = 0. For CUST this returns the the value of **DAsInPrinting** (0 or 1)

SetIntegerRes sets the value of the integer in the WhichInteger position, in the resource of type ResType with ID = ResID.

For example:

SetIntegerRes("CUST";0;3;16) will set the value of the 3rd integer in the CUST resource with ID = 0 to 16. This sets the **CursorSpinRate** to 16.

1. "4D TMPLS" file containing ResEdit Templates for 4th Dimension.



2. External procedures Note on resource manipulation and "Proc.ext" containing **GetIntegerRes** and **SetIntegerRes**.

