

## The Fox Hunt

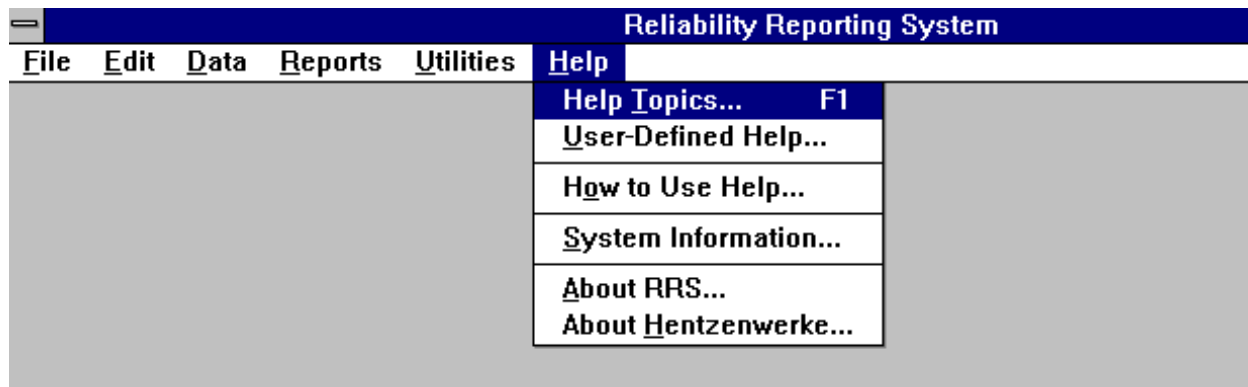
Help your users help themselves with User-Defined Help

By Whil Hentzen

Most systems include a healthy dose of “hidden knowledge” that users don’t communicate to the developer - either through ignorance or forgetfulness. In many cases, this knowledge isn’t necessary for the development of the system, but it is needed in order to use the system. This knowledge often takes the forms of company procedures that may change or evolve as the system is used.

Most times, this hidden knowledge is passed on from user to user much like folk tales of old - impromptu meetings at the water cooler or brief training sessions when a new user is added to the system. Occasionally, however, someone makes an effort to document these procedures in the form of memos or even a “manual.” The trouble, of course, is that that paper trail is never around when the user needs it, and instead of looking for it, they’ll just make a guess at what to do.

A handy feature to include with your applications is the ability for users to add their own notes to the application - as they’re using it. This way, the hidden knowledge gets captured at the instant it’s realized, and by including with the app, it’s available to all users at any time.



Providing User-Defined Help is easy to do with a stand-alone function. First, you’ll need three files: SYS\_UDH.DBF, SYS\_UDH.FPT, and SYS\_UDH.CDX.

```
Structure for table:      f:\mmc\rrs\source\sys_udh.dbf
Field  Field Name  Type      Width  Dec  Index  Collate
  1   CTOPIC      Character  10          Asc  Machine
  2   CTOPIC2DIS  Character  30          Asc  Machine
  3   MDETAILS    Memo       10
  4   CCLASS     Character  20
  5   NORDER     Numeric    3          Asc  Machine
** Total **                74
```

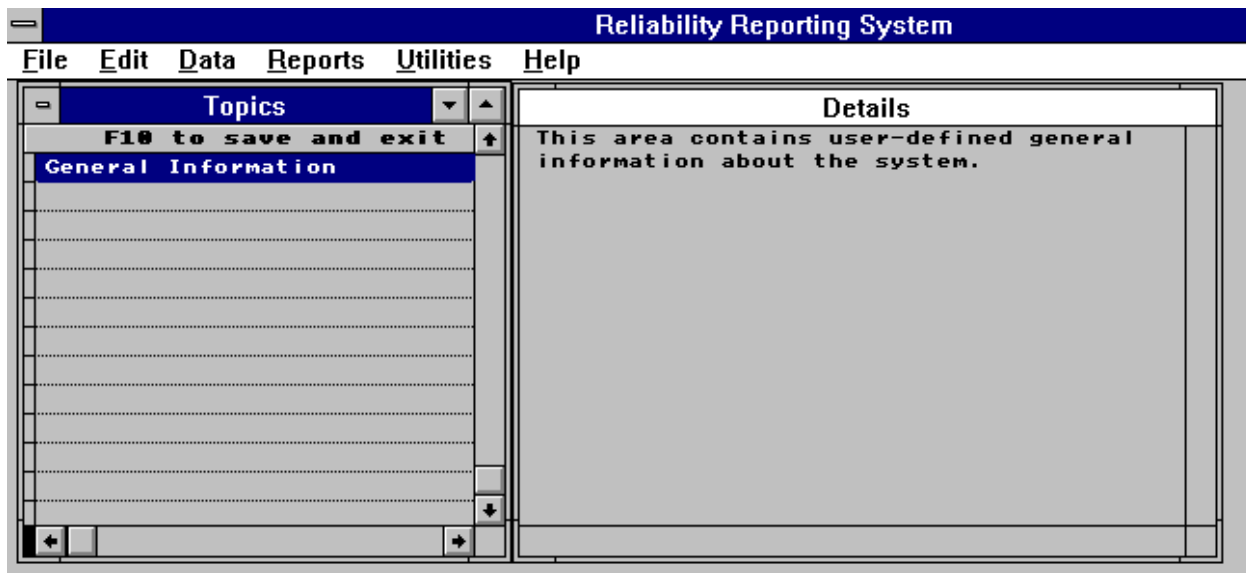
The “cTopic” field contains the name of the current window, the “cTopic2Dis” field contains the English description of the window, the “mDetails” field contains the text that the user enters for this particular screen, and the “cClass” and “nOrder” fields are not currently used. I’ve put them in the table in anticipation of future enhancements.

Next, include a “User-Defined Help” bar on the Help menu, and have it call the program F\_UDHELP.PRG.

When the user executes the function, the program opens up SYS\_UDH and sets order to cTopic. You could use the “nOrder” field instead, and provide the user the ability to change the contents of this field so they can control the order of the topics themselves.

The trick to this program is using the name of WONTOP to determine where in the application the user is, and finding the related help record, if there is one. If there isn’t, the program asks the user if they want to create one on the fly. Note that I ship SYS\_UDH with a single empty record that will display when the user selects User Defined Help when at the main menu. Naturally, the program traps for that instance where, for some reason, that record is gone.

Once the appropriate help record is found or created, a pair of side-by-side windows are opened - the left window with the available help topics, and the right window containing the text for that topic.



```

*      Program: F_UDHELP.PRG
*      Purpose: Provides user-defined help
*      Version: 1.0
*      Platforms: FoxPro 2.5a/D/W
*      Syntax: do F_UDHELP
*      Params: none
*      Return: nothing
*      Samples:
*      Files: SYS_UDH.DBF/FPT/CDX
*      Required:
*      Files: None
*      Created:
*      Notes:

private m.i, m.jcCurWA

m.jcCurWA = alia()
select 0
use SYS_UDH order cTopic

*
* look for the name of the WONTOP in the topic list
*
do case
case empty(wontop()) or upper(wontop()) = "VIEW"
*
* the user is in the main menu, or
* the developer has the view window open
*
if seek("MAIN MENU")
*
* MAIN MENU is generally shipped with the SYS_UDH table
*
=f_udhwin()
else
*
* just in case the developer forgot...
*
wait wind "User-Defined Help not found for the main menu. Create help (Y/N)?" to i
if upper(i) = "Y"
m.cTopic = upper(wontop())
m.cTopic2Dis = "Enter your topic description here"

```

```

        insert into (dbf()) from memvar
        =f_udhelp()
    endif
endif
case seek(upper(wontop()))
*
* we have found the name of the screen that the user is in
*
    =f_udhwin()
case !seek(upper(wontop()))
*
* the name of this screen is not in SYS_UDH
*
wait window "User-Defined Help not found for this screen. Create help (Y/N)?" to i
if upper(i) = "Y"
    m.cTopic = upper(wontop())
    m.cTopic2Dis = "Enter your topic description here"
    insert into (dbf()) from memvar
    =f_udhwin()
endif
otherwise
    wait window "DEV: we were seeking on WONTOP but no conditions were met"
endcase
if !empty(m.jcCurWA)
    select &jcCurWA
else
    select 0
endif
return .t.

function f_udhwin
define window w_br0 from 0,-1 to srows()-1,79
define window w_br1 in window w_br0 ;
    system noclose float grow mini zoom ;
    from 0,0 to srows()-2,32 ;
    title "Topics"
define window w_br2 in window w_br0 ;
    system noclose float grow mini zoom ;
    from 0,33 to srows()-2,79 ;
    title "Details"
on key label F10 do f_clossave
modify memo MDETAILS window w_br2 nowa
browse ;
nomenu ;
fields ;
    CTOPIC2DIS :H=" F10 to save and exit", ;
    MDETAILS:W=.F. ;
window w_br1 ;
color scheme 10
on key label F10
return .t.

function f_clossave
use in SYS_UDH
release window w_br0
release window w_br1
release window w_br2
return .t.

```

The best part of this routine is that you can simply “plug it in” to your applications and forget about it. It’s self-maintaining and requires no additional programming to include it in new systems.