

# Make.List Subroutine

## Description

The Make.List subroutine creates an active select list of keys from the passed list of keys.

## Syntax

**Make.List** (*cursor*, *keylist*, *f\_datavar*, *f\_dictvar*)

## Parameters

The Make.List subroutine has the following parameters:

Parameter	Description
<i>Cursor</i>	Specify the cursor number. Valid cursors numbers are 0 through 8.
<i>Keylist</i>	An @fm-delimited list of keys. <b>Note: The Keylist parameter is passed by reference and the Make.List subroutine will clear the variable passed.</b>
<i>f_datavar</i>	The file handle for the data file as set by the open statement.
<i>f_dictvar</i>	The file handle for the dictionary file as set by the open statement.

Cursor 0 is the default cursor and is the only cursor used by RList and the simple forms of the Select and Readnext statements.

## See also

[Activate\\_Save\\_Select](#), [ReadNext](#), [RList](#), [Select](#), [Save\\_Select](#)

## Example

```

subroutine make_list_example(parent)

Declare Function Unassigned, Get_Status, Get_Property
Declare Subroutine Msg, MAKE.LIST, fsmsg, Set_Status, Save_Select

$Insert Logical
$Insert Msg_Equates

* Initialize Variables
errCodes = ""
errFlag = ""

* Determine Parent Window
If Unassigned(parent) Then parent = @Window
frame = Get_Property(parent,"MDIFRAME")
If frame Then parent = frame

* Open Files
Open "CUSTOMER" To f_customers Then
    Open "DICT.CUSTOMERS" To d_customers Else
        errText = "Unable to open the Customers Dictionary file."
        errFlag = TRUE$
    End
End Else
    errText = "Unable to open Customers file."
    errFlag = TRUE$
End

If errFlag Else

    * A List of Customer keys
    custList = "1" : @vm : "2" : @vm : "22" : @vm : "26"

    * Create the List in Cursor 0 and if successful save the list
    Set_Status(0)
    MAKE.LIST(0, custList, f_customers, d_customers)
    If Get_Status(errCodes) Then
        FSMsg(errCodes)
    End Else
        Set_Status(0)
        Save_Select("", "MAKE_LIST_TEST", "", "")
        If Get_Status(errCodes) then
            FSMsg(errCodes)
        end
    End
End

if errFlag Then
    msgStruct = ""
    msgStruct<MCAPTION$> = "Error"
    msgStruct<MICON$> = "H"
    msgStruct<MTEXT$> = errText : " |Error Code: " : @file.error<1>
    Msg(Parent,msgStruct)
end

return

```