Dial: Dial [dtmf, phonefield, hangup, switch, dbn (first, last, selected)]

```
Set Variable [ $lltemp; Value: Let( [
    $sp = LeftWords( Get ( ScriptParameter) ; 1);
    $sp2 = Middle ( Get ( ScriptParameter) ; Length( $sp) + 2 ; 40 ) ;  //use middel because of # key
    $line = If( window.user::xPrefUser_PhoneLine < 1 ; 1;window.user::xPrefUser_PhoneLine) ;
    $type = window.user::xPrefUser_PhoneType[$line];
    $wn = Get ( WindowName ) ;
    $timer  = window::xSys_PhoneTimerFlag ;
    $sfield = Get ( ActiveFieldTableName ) & "::" & Get ( ActiveFieldName ) ;
    $sstart = Get ( ActiveSelectionStart ) ;
    $slength = Get ( ActiveSelectionSize )
    ] ;
    window::xPt_ItemIDForScript
    ) ]
#
If [ $sp = "switch" ]
    Show Custom Dialog [ Message: "Current line "  & window.user::xPrefUser_PhoneLine & " is selected.¶¶"  &

        "Line 1  " & window.user::xPrefUser_PhoneCounter[1] & " calls           " & window.user::xPrefUser_PhoneCounterDay[1]  & "¶"
        &
        "Line 2  " & window.user::xPrefUser_PhoneCounter[2] & " calls           " & window.user::xPrefUser_PhoneCounterDay[2]  & "¶"
        &
        "Line 3  " & window.user::xPrefUser_PhoneCounter[3] & " calls           " & window.user::xPrefUser_PhoneCounterDay[3]  & "¶"
        &
        "¶" & "Switch to Line..."; Buttons: “OK”, “Cancel” ]
    Set Field [ window.user::xPrefUser_PhoneLine; Case(
        Get ( LastMessageChoice ) = 1; 1 ;
        Get ( LastMessageChoice ) = 2; 2 ;
        3  ) ]
    Exit Script [  ]
End If
#
If [ $sp = "dbn" ]
    Set Variable [ $name; Value:Case(
        $sp2 = "first" ; window.contact::First Name ;
        $sp2 = "last" ; window.contact::Last Name ;
        $sp2 = "selected" ; Middle ( Get ( ActiveFieldContents ) ; Get ( ActiveSelectionStart ) ; Get ( ActiveSelectionSize ) )
        ) ]
    Loop
        Exit Loop If [ Let( $count = $count + 1 ; $count)   >  10 ]
        Set Field [ window::xSys_DBN[$count]; Let( aa = Middle( Lower( $name ) ; $count ; 1) ;
            Case(
            Length( Filter ( aa ; "abc" ) ) > 0 ;  2 ;
            Length( Filter ( aa ; "def" ) ) > 0 ;  3 ;
            Length( Filter ( aa ; "ghi" ) ) > 0 ;  4 ;
            Length( Filter ( aa ; "jkl" ) ) > 0 ;  5 ;
            Length( Filter ( aa ; "mno" ) ) > 0 ;  6 ;
            Length( Filter ( aa ; "pqrs" ) ) > 0 ;  7 ;
            Length( Filter ( aa ; "tuv" ) ) > 0 ;  8  ;
            Length( Filter ( aa ; "wxyz" ) ) > 0 ;  9
            )
            ) ]
    End Loop
    Go to Object [ Object Name: Case (
        $sp2 = "first" ; "tab.dtmf.dbnfirst";
        $sp2 = "last" ; "tab.dtmf.dbnlast";
        $sp2 = "selected" ; "tab.dtmf.dbnselected"
        ) ]
    Exit Script [  ]
End If
#
#Set phone number
If [ $sp = "sidebar" ]
    Set Field [ window::xPt_ItemIDForScript; portal.sidebar::xSidebar_IDitemNumberInternal ]
    Set Variable [ $phonenumber; Value:Left( Filter ( item.contact::Phone Area Code ; "0123456789") ; 3 ) &  Left(  Filter ( item.contact::
        Phone; "0123456789") ; 7 ) ]
    Set Variable [ $$ext; Value:Trim ( Middle( item.contact::Phone  ; 9; 20 ) ) ]
Else If [ $sp = "phone1" ]
    Set Variable [ $phonenumber; Value:Left( Filter ( window.contact::Phone Area Code ; "0123456789") ; 3 ) &  Left(  Filter ( window.
        contact::Phone ; "0123456789") ; 7 ) ]
    Set Variable [ $$ext; Value:Trim ( Middle( window.contact::Phone  ; 9; 20 ) ) ]
Else If [ $sp = "phone2" ]
    Set Variable [ $phonenumber; Value:Left( Filter ( window.contact::Phone2 Area Code ; "0123456789") ; 3 ) &  Left(  Filter ( window.
        contact::Phone2; "0123456789") ; 7 ) ]
    Set Variable [ $$ext; Value:Trim ( Middle( window.contact::Phone2  ; 9; 20 ) ) ]
Else If [ $sp = "selected" ]
    Set Variable [ $phonenumber; Value:Left( Filter (  Middle( Get ( ActiveFieldContents ) ; Get ( ActiveSelectionStart ) ; Get
        ( ActiveSelectionSize ) )  ; "0123456789") ; 10 ) ]
    Set Variable [ $$ext; Value:Let(
        $a = Trim (  Middle( Get ( ActiveFieldContents ) ; Get ( ActiveSelectionStart ) ; Get ( ActiveSelectionSize ) ) )
        ;
        Case(
        Length( $a ) > 9 ; Trim( Middle( $a; 13; 20) ) ;
        $a
        )
        ) ]
End If
#
#DTMF
If [ $sp = "dtmf" ]
    If [ $type = "skype" ]
        Set Variable [ $dtmf; Value:Case(
            $sp2 = "s" ;   Middle( Get ( ActiveFieldContents ) ; Get ( ActiveSelectionStart ) ; Get ( ActiveSelectionSize ) ) ;
            $sp2 = "e" ;  $$ext ;
            $sp2
            ) ]
        Set Variable [ $end; Value:Let(   [
            $dtmf = Filter ( $dtmf  ; "0123456789-p*#,. " );
            $count = 0
            ];
            Length($dtmf)  ) ]
        Loop
            Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
            Set Variable [ $key; Value: Middle($dtmf; $count ; 1 ) ]
            If [ $key = "p" or
                $key = " " or
                $key = "," or
                $key = "." or
                $key = "-" ]
                Pause/Resume Script [ Duration (seconds): If( Length( $dtmf) = 1; 0; 1.3) ]
            Else
                If [ window::xSys_PhoneTest = 1 ]
                    Beep
                Else
                    Perform AppleScript [ Calculated AppleScript: "tell application \"Skype\" ¶"  &
                              "set calls to send command \"SEARCH ACTIVECALLS\" script name \"My Script\" ¶"  &
                              "set callID to last word of calls ¶"  &
                              "send command \"SET CALL \" & callID & \" DTMF "   &  $key  & "\" script name \"My Script\" ¶"  &
                         "end tell" ]
                End If
                Pause/Resume Script [ Duration (seconds): Case(
                    Length( $dtmf) = 1; 0;
                    $count = Length( $dtmf) ; 0;
                    .6) ]
            End If
        End Loop
    End If
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Exit Script [  ]
End If
#
Set Field [ window::xSys_PhoneHangup; "" ]
```

```
#Hangup
If [ $sp = "Hangup" ]
    Go to Object [ Object Name: "tab.dtmf.blank" ]
    Go to Object [ Object Name: "tab.dial.dial" ]
    If [ window::xSys_PhoneTest = 1 ]
        Beep
    Else
        If [ $type = "Skype" ]
            Perform AppleScript [ Native AppleScript: tell application "Skype"
                        set calls to send command "SEARCH ACTIVECALLS" script name "My Script"
                        set callID to last word of calls
                        send command "ALTER CALL " & callID & " HANGUP" script name "My Script"
                end tell ]
        Else If [ $type = "ExpressTalk" ]
            Perform AppleScript [ Calculated AppleScript: "tell application \"ExpressTalk\"¶
                        activate¶"  &

                "tell application \"System Events\" to keystroke \"1\" using {command down}¶"  &

                "end tell" ]
        End If
        Perform AppleScript [ Native AppleScript: tell application "Filemaker Pro"
                activate
            end tell
             ]
    End If
    Set Field [ window.system::xSys_Pass.DialInitials; Substitute ( window.system::xSys_Pass.DialInitials ; window::xxIDL_Owner &
        "¶"  ; "" ) ]
    Set Field [ window::xPt_ItemIDForScript; $lltemp ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Exit Script [  ]
End If
#
Set Field [ window::xSys_PhoneHangup; 1 ]
Go to Object [ Object Name: "tab.dtmf.main" ]
Go to Object [ Object Name: "tab.dial.hangup" ]
#
If [ window::xSys_PhoneTest = 1 ]
Else
    #counter
    Set Field [ window.user::xPrefUser_PhoneCounter[If( $line > 1;  $line ; 1 )]; Case(
        window.user::xPrefUser_PhoneCounterDay[$line] = Get(CurrentDate); window.user::xPrefUser_PhoneCounter[$line] + 1 ;
        1
        ) ]
    Set Field [ window.user::xPrefUser_PhoneCounterDay[If( $line > 1;  $line ; 1 )]; Get(CurrentDate) ]
    If [ window.user::xPrefUser_PhoneCounter[$line] = window.user::xPrefUser_PhoneMAXCalls ]
        Show Custom Dialog [ Message: window.user::xPrefUser_PhoneCounter[$line] & " calls reached on line "  & window.user::
            xPrefUser_PhoneLine & ".  Switch line and redial?"; Buttons: “Line 1”, “Line 2”, “Line 3” ]
        Set Field [ window.user::xPrefUser_PhoneLine; Case(
            Get ( LastMessageChoice ) = 1; 1 ;
            Get ( LastMessageChoice ) = 2; 2 ;
            3  ) ]
        Set Field [ window::xPt_ItemIDForScript; $lltemp ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
        Exit Script [  ]
        Go to Object [  ]
        Set Selection [  ]
    End If
    #
    #Set Dialed fields
    Set Field [ window.system::xSys_Pass.DialInitials; window::xxIDL_Owner & "¶" ]
    Set Field [ window.contact::Call Last Attempt; Get(CurrentDate) ]
    #
    #DIAL
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    If [ $type = "Skype" ]
        Perform AppleScript [ Native AppleScript: do shell script "sudo renice -20 -p `ps -ax l grep '[/]Skype' l awk '{print $1}'`" ]
        Set Variable [ $phonenumber; Value:"+1" & $phonenumber ]
        Perform AppleScript [ Calculated AppleScript: "tell application \"Skype\" ¶"  &
                "send command \"CALL " & $phonenumber   & "\" script name \"My Script\" ¶"  &
            "end tell" ]
        If [ 1 = 2 ]
            Open URL [ "callto://"  &   $phonenumber ]
                [ No dialog ]
        End If
    Else If [ $type = "ExpressTalk" ]
        Set Variable [ $phonenumber; Value:"1" & $phonenumber ]
        Perform AppleScript [ Calculated AppleScript: "tell application \"ExpressTalk\"¶
                activate¶"  &

            "tell application \"System Events\" to keystroke tab¶"  &
            "tell application \"System Events\" to keystroke \"c\" using {command down}¶
            tell application \"System Events\" to keystroke \"" & $phonenumber & "\"¶" &

            "tell application \"System Events\" to keystroke tab¶"  &
            "tell application \"System Events\" to keystroke \"c\" using {command down}¶
            tell application \"System Events\" to keystroke \"" & $phonenumber & "\"¶" &

            "tell application \"System Events\" to keystroke tab¶"  &

            "tell application \"System Events\" to keystroke \"1\" using {command down}¶"  &

            "end tell" ]
    End If
    Set Field [ window::xSys_PhoneHangup; 1 ]
    Go to Object [ Object Name: "tab.dtmf.main" ]
    Go to Object [ Object Name: "tab.dial.hangup" ]
    #
    #RECORD
    If [ window.user::xPrefUser_PhoneRecordChoice[$line] = 1 ]
        Perform AppleScript [ Native AppleScript: delay .6
            tell application "Skype"
                set calls to send command "SEARCH ACTIVECALLS" script name "My Script"
                set callID to last word of calls
                send command "ALTER CALL " & callID & " SET_OUTPUT file=" & quote & "temp.wav" & quote & ",SOUNDCARD=" &
            quote & "default" & quote script name "My Script"
            end tell
             ]
        Perform Script [ “WindowNew [ main, folder, folder options, pref, note, newemail, importemail, phonerecord, websites]”;
            Parameter: "phonerecord" ]
    End If
End If
#
Set Field [ window::xPt_ItemIDForScript; $lltemp ]
Perform AppleScript [ Native AppleScript: delay 1.1

    tell application "Filemaker Pro"
        activate
    end tell
     ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Exit Script [  ]
#
```

```
Perform AppleScript [ Native AppleScript: delay .8

    tell application "System Events"
    set visible of process "Skype" to false
    end tell


    tell application "Filemaker Pro"
        activate
    end tell
     ]
Show Custom Dialog [ Message: "sp: "  & $sp  & "¶" &
    "sp2: "  & $sp2  & "¶" &
    "phone#: "  & $phonenumber  & "¶" &
    "type: "  & $type  & "¶" &
    "line: "  & $line  & "¶" &
    "dtmf: "  & $dtmf  & "¶" &
    "line: "  & $line  & "¶" &
    "wn: "  & $wn  & "¶" &
    "timer: "  & $timer  & "¶"; Buttons: “OK”, “Cancel” ]
If [ Get(LastMessageChoice) = 2 ]
    Exit Script [  ]
End If
Perform AppleScript [ Native AppleScript: tell application "Skype"
        set calls to send command "SEARCH ACTIVECALLS" script name "My Script"
        set callID to last word of calls
        send command "ALTER CALL " & callID & " HANGUP" script name "My Script"
    end tell
     ]
#https://developer.skype.com/Docs/ApiDoc/Commands
```

Dial: Dial Phone 1

Go to Object [Object Name: "button.phone1"]
Pause/Resume Script [Duration (seconds): .2]
Perform Script [“Dial [dtmf, phonefield, hangup, switch, dbn (first, last, selected)]”; Parameter: "phone1"]

Dial: Dial Phone 2

```
Go to Object [ Object Name: "button.phone2" ]
Pause/Resume Script [ Duration (seconds): .2 ]
Perform Script [ “Dial [ dtmf, phonefield, hangup, switch, dbn (first, last, selected) ]”; Parameter: "phone2" ]
```

Dial: Dial Selected

```
Go to Object [ Object Name: "button.selected" ]
Pause/Resume Script [ Duration (seconds): .2 ]
Perform Script [ “Dial [ dtmf, phonefield, hangup, switch, dbn (first, last, selected) ]”; Parameter: "selected" ]
```

Dial: export report for jenna

#goto data.note
find for commun, vm, missed, in sys_type
then find for date and kmc
then export, company, name, etc from primary contact
sort by company

Open and maintenance: Open Script

```
Adjust Window
      [ Hide ]
Go to Layout [ “Main Window” (window) ]
Show/Hide Status Area
      [ Hide ]
Show/Hide Text Ruler
      [ Hide ]
New Record/Request
If [ IsEmpty(window.user::xPrefUser_MainWindowSize) ]
      Perform Script [ “WindowResize [centered, full, ]”; Parameter: "full" ]
Else
      Move/Resize Window [ Current Window; Height: window.user::xPrefUser_MainWindowSize[1]; Width: window.user::
            xPrefUser_MainWindowSize[2]; Top: window.user::xPrefUser_MainWindowSize[3]; Left: window.user::
            xPrefUser_MainWindowSize[4] ]
End If
Refresh Window
Freeze Window
#
Perform Script [ “Daily Reminders” ]
Set Variable [ $$as; Value:"tell application \"FileMaker Pro\" to set cellValue of repetition 1 of cell \"clipboard\" of table \"system\"  of
database \"dddddd\" to \"\"" & "¶" &
      "set someVariable to \"\"" & "¶" &

      "try¶" &

      "set someVariable to do shell script \"cccccc\" user name \"aaaaaa\" password \"pppppp\" with administrator privileges¶" &

      "on error errorMsg¶" &
            "set someVariable to \"error \"  & errorMsg ¶" &

      "end try¶" &

      "if someVariable is \"error\" then¶" &
      "else¶" &
      "end if¶" &

      "tell application \"FileMaker Pro\" to set cellValue of repetition 1 of cell \"clipboard\" of table \"system\"  of database \"dddddd\" to
      someVariable¶"

      //"display dialog someVariable¶"
       ]
#
Set Variable [ $a; Value:Let( [
      $$drop.note = "" ;
      $$drag.ID = "" ;
      $$drop.ID = "" ;
      $$drag.portal = "";
      $$drag.note = "";
      $$dragheadername = "";
      $$drop.sidebarID = "" ;
      $$drag.sidebarID = ""  ;
      $$Drag.foldertype = "" ;
      $$Draglist = "" ;
      $$Dontchange = "" ;
      $$EditLink = "" ;
      $$ClickTimer = ""
      ] ; "" ) ]
Perform Script [ “FolderSelect [ link, load, save ] ”; Parameter: $$inboxID ]
Set Variable [ $link; Value:window::xPt_SelectedItems__CurrentClickedLink ]
Set Field [ window::xPt_ItemIDForScript; $link ]
Set Field [ window::xPt_ContactID; join::xxIDL_LinkPrimaryItem ]
Set Field [ window::xPt_ItemIDForScript; "" ]
Set Variable [ $a; Value:Let(  [
      $count = 0 ;
      $end = 5
      ] ; ""
      ) ]
Loop
```

```
    Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
    Set Field [ window.user::xPrefUser_PhoneCounter[$count]; Case(
        window.user::xPrefUser_PhoneCounterDay[ $count ] = Get(CurrentDate); window.user::xPrefUser_PhoneCounter[ $count ] ;
        ""
        ) ]
    Set Field [ window.user::xPrefUser_PhoneCounterDay[$count]; Case(
        window.user::xPrefUser_PhoneCounterDay[ $count ] = Get(CurrentDate); Get(CurrentDate)  ;
        ""
        ) ]
End Loop
Perform Script [ “Timer” ]
Install OnTimer Script [ “Timer”; Interval: 30 ]
If [ window.user::xPrefUser_PhoneLine  ≠  1 ]
    Refresh Window
    Show Custom Dialog [ Message: "Dialer set to line "  & window.user::xPrefUser_PhoneLine & ".  Switch line?"; Buttons: “Line 1”,
        “Line 2”, “Line 3” ]
    Set Field [ window.user::xPrefUser_PhoneLine; Case(
        Get ( LastMessageChoice ) = 1; 1 ;
        Get ( LastMessageChoice ) = 2; 2 ;
        3  ) ]
End If
If [ window.user::xSys_ServerName = Get ( HostName ) ]
    New Window [ Name: "~Temp~"; Height: 80; Width: 80; Top: -400; Left: 20 ]
    Pause/Resume Script [ Duration (seconds): .3 ]
    Close Window [ Current Window ]
    Perform AppleScript [ Native AppleScript: do shell script "sudo renice -18 -p `ps -ax | grep '[/]FileMaker' | awk '{print $1}'`" ]
    #EDITOR=pico sudo visudo
    %staff ALL=NOPASSWD: /usr/bin/renice
    Open File [ “orders” ]
        [ Open hidden ]
    Close File [ “orders” ]
    Open File [ “quotes” ]
        [ Open hidden ]
    Close File [ “quotes” ]
End If
#
#
Exit Script [ Result: Let( [
    $$drop.note = "" ;
    $$drag.ID = "" ;
    $$drop.ID = "" ;
    $$drag.portal = "";
    $$drag.note = "";
    $$dragheadername = "";
    $$drop.sidebarID = "" ;
    $$drag.sidebarID = ""  ;
    $$Drag.foldertype = "" ;
    $$Draglist = "" ;
    $$Dontchange = ""
    ] ; "" ) ]
```

Open and maintenance: New Script

```
Show Custom Dialog [ Message: window.user::xSys_ServerName  & "¶" &  Get ( HostName ); Buttons: “OK”, “Cancel” ]
```

Open and maintenance: renice

```
Perform AppleScript [ Native AppleScript: do shell script "sudo renice -20 -p `ps -ax | grep '[/]Skype' | awk '{print $1}'`" ]
```

Open and maintenance: renice Copy

```
Perform AppleScript [ Native AppleScript: do shell script "sudo renice -20 -p `ps -ax | grep '[/]FileMaker' | awk '{print $1}'`" ]
Perform AppleScript [ Native AppleScript: do shell script "sudo renice +20 -p `ps -ax | grep '[/]configd' | awk '{print $1}'`" ]
Perform AppleScript [ Native AppleScript: do shell script "sudo renice +20 -p `ps -ax | grep '[/]Finder' | awk '{print $1}'`" ]
Perform AppleScript [ Native AppleScript: do shell script "sudo renice +20 -p `ps -ax | grep '[/]kernel' | awk '{print $1}'`" ]
Perform AppleScript [ Native AppleScript: do shell script "sudo renice +20 -p `ps -ax | grep '[/]Preview' | awk '{print $1}'`" ]
Perform AppleScript [ Native AppleScript: do shell script "sudo renice +20 -p `ps -ax | grep '[/]WindowServer' | awk '{print $1}'`" ]
```

Open and maintenance: Close Script

```
Set Variable [ $$FolderClickTimer; Value:Get(CurrentTime) - 3 ]
Go to Layout [ “Main Window” (window) ]
Enter Browse Mode
Perform Script [ “FolderSelect [ link, load, save ] ”; Parameter: window::xID_FolderSelectedID & " save" ]
```

Open and maintenance: Daily Reminders

```
#
#make new system folders if not existing, someday
#
Set Variable [ $a; Value:Let( [
    $$drop.note = "" ;
    $$drag.ID = "" ;
    $$drag.PortalID = "" ;
    $$drop.ID = "" ;
    $$drag.portal = "";
    $$drag.note = "";
    $$dragheadername = "";
    $$drop.sidebarID = "" ;
    $$drag.sidebarID = ""  ;
    $$Drag.foldertype = "" ;
    $$Draglist = "" ;
    $$DraglistPortalID = "" ;
    $$Dontchange = ""
    ] ; "" ) ]
Set Variable [ $a; Value:Let( [
    $$DeleteKeyFolder = window.user::xFolder_SystemDeleteKeyFolder ;
    $$DeleteKeyItem = window.user::xFolder_SystemDeleteKeyItem ;
    $$DeleteKeyOrphaned = window.user::xFolder_SystemDeleteKeyOrphaned
    ] ; "" ) ]
#
Set Field [ window::xID_FolderHeaderTypeExclude; "LISTS¶FOLDERS¶USERS" ]
Set Variable [ $a; Value:Let(  [
    $count = 0 ;
    $$InboxID = "" ;
    $$TrashID = "";
    $$RemovedID =  "";
    $$FutureID =  "" ;
    $LL = List( portal.sidebar.folder::xxIDL_IDNumberInternal );
    $end = WordCount( $LL )
    ] ; ""
    ) ]
Loop
    Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
    Set Field [ window::xPt_ItemIDForScript; MiddleWords($LL;$count;1) ]
    Set Variable [ $a; Value:Let( [
        $$InboxID = If( item.folder::Folder Type = "Inbox"; item.folder::xxIDL_IDNumberInternal; $$InboxID ) ;
        $$TrashID =  If( item.folder::Folder Type = "Trash"; item.folder::xxIDL_IDNumberInternal; $$TrashID );
        $$FutureID =  If( item.folder::Folder Type = "Future"; item.folder::xxIDL_IDNumberInternal; $$FutureID )
        ];"" ) ]
End Loop
Set Field [ window::xID_FolderHeaderTypeExclude; window.user::xPrefUser_FolderHeaderTypeExclude ]
#
Set Field [ window.system::xxIDL_SystemRecord; 1 ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
If [ window.system::xSys_SystemDate  ≠  Get(CurrentDate) ]
    New Window [ Name: "~temp~"; Height: 1; Width: 1; Top: -350 ]
    Go to Layout [ “data.window” (window) ]
    New Record/Request
    #set reminder dots
    Set Variable [ $a; Value:Let(  [
        $count = 0 ;
        $movelist = "" ;
        $moveIDlist = "" ;
        $coyplist = "" ;
        $LL = List( window.dailyreminder::xxIDL_LinkList );
        $end = WordCount( $LL )
        ] ; ""
        ) ]
    Loop
        Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
        Set Field [ window::xPt_ItemIDForScript; MiddleWords($LL;$count;1) ]
        Set Field [ item.reminder::Flag; Case(
                item.reminder::xFolder_ItemResetDailyPref = 1;
                Case (
                item.reminder::Flag = "2"; "2" ;
                item.reminder::Flag = "0"; "0" ;
                 "1" ) ;
                item.reminder::Flag
                ) ]
            Set Field [ item.reminder::Time; Get(CurrentTime) ]
            Set Variable [ $movelist; Value:If(
                item.reminder::xFolder_ItemMoveToInboxPref = 1;
                item.reminder::xxIDL_LinkList ;
                "" ) ]
            Set Variable [ $moveIDlist; Value:If(
                item.reminder::xFolder_ItemMoveToInboxPref = 1;
                item.reminder::xFolder_ItemMoveToInboxPrefIDNumbers  ;
                "" ) ]
            Set Variable [ $copylist; Value:If(
                item.reminder::xFolder_ItemMoveToInboxPref   or window.user::xFolder_SystemCopyToInbox;
                $copylist & "¶" & item.reminder::xxIDL_LinkList ;
                $copylist ) ]
            Commit Records/Requests
                [ Skip data entry validation; No dialog ]
            Perform Script [ “Move/Copy/ClearToAnyFolder [ $$drop.ID [1], $$DragList [2] cleans, g(sp) = from folders ] ”; Parameter: Let
                ( [ $$drop.id = $$InboxID ; $$DragList = $movelist ] ; $moveIDlist ) ]
        End Loop
        Perform Script [ “Move/Copy/ClearToAnyFolder [ $$drop.ID [1], $$DragList [2] cleans, g(sp) = from folders ] ”; Parameter: Let( [ $
            $drop.id = $$InboxID ; $$DragList = $copylist ] ; "" ) ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
        Delete Record/Request
            [ No dialog ]
        Close Window [ Current Window ]
End If
#
Set Field [ window.system::xSys_SystemDate; Get(CurrentDate) ]
Set Variable [ $$systemdate; Value:Get(CurrentDate) ]
Set Field [ window::xPt_ItemIDForScript; "" ]
#
Exit Script [ Result: Let( [
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drag.PortalID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$DraglistPortalID = "" ;
        $$Dontchange = ""
        ] ; "" ) ]
```

Re-Login [ ]

Open and maintenance: Search Directories [card]

```
If [ Get(ScriptParameter ) = "linkedin" ]
Open URL [ "http://www.linkedin.com/search/fpsearch?fname="  & window.contact::First Name & "&lname=" & window.contact::Last
Name &
        If( not IsEmpty( window.contact::Ship Zip ) ; "&searchLocationType=I&countryCode=us&postalCode="  & Left( window.contact::Ship
        Zip ;5)  & "&distance=50" ; "")  &
        "&page_num=1&search=&pplSearchOrigin=MDYS&viewCriteria=1&sortCriteria=R&redir=redir"

        /* lname=Donnelly&company=microsoft&currentCompany=CP */ ]
        [ No dialog ]
Else If [ Get(ScriptParameter ) = "jigsaw" ]
Open URL [ "http://www.jigsaw.com/SearchAcrossCompanies.xhtml?
&opCode=search&orderby=0&order=0&nextURL=&mode=0&countryvalue=&statevalue=&metrovalue=&indvalue=&subindvalue=&showAd
vancedSearch=true&cnTitle=&cmName="   & window.contact::Company Name &   "&cnEmail=&cnName="  & window.contact::Last Name
&
        "&cnFirstName="   & window.contact::First Name &
        "&cnCountry=9000&cnArea1=&cnZip=&cmFortune=0&screenNameType=0&screenName=" ]
        [ No dialog ]
Exit Script [  ]
Open URL [ "http://www.jigsaw.com/FreeTextSearch.xhtml?opCode=search&autoSuggested=true&freeText="  &  window.contact::
        Company Name ]
        [ No dialog ]
Else If [ Get(ScriptParameter ) = "searchmaps" ]
Open URL [ "http://maps.google.com/maps?" &
        "q=" &
        window.contact::Company Name  & ", "  & window.contact::Ship City  & ", " & window.contact::Ship State

        /*

        /*Address=*/ window.contact::Ship Address & "," & /*City=*/ window.contact::Ship City & "," & /*State=*/ window.contact::Ship State &
        "," & /*Zip Code=*/ window.contact::Ship Zip & "," & /*Country=*/ ""


        */ ]
        [ No dialog ]
Else If [ Get(ScriptParameter ) = "searchcompany" ]
Open URL [ Let(
        a = window.contact::Company Name  & ", "  & window.contact::Ship City  & ", " & window.contact::Ship State  ;
        Substitute ( window.contact.user::xPrefSys_SearchCompanyURL ; "#query#" ; a )
        )

        //"http://www.google.com/search?" & "q=" & window.contact::Company Name  & ", "  & window.contact::Ship City ]
        [ No dialog ]
End If
If [ Get(ScriptParameter) = "card" ]
        Set Variable [ $link; Value:window::xPt_ContactID ]
        Commit Records/Requests
                [ Skip data entry validation; No dialog ]
        New Window [ Height: 1; Width: 1; Top: -350 ]
        Go to Layout [ “Research Card” (window) ]
        Show/Hide Status Area
                [ Hide ]
        New Record/Request
        Set Field [ window::xPt_ContactID; $link ]
        Set Field [ window::xID_RelatedWindow; $relatedwindow ]
        Set Field [ window::xPt_MainListLayout; "ResearchCard" ]
        Commit Records/Requests
                [ Skip data entry validation; No dialog ]
        Perform Script [ “ItemTab” ]
        Move/Resize Window [ Current Window; Height: 780; Width: 295; Top: 2; Left: 2 ]
End If
```

Open and maintenance: MiscScripts [...]

```
Set Variable [ $a; Value:Let( [
    $ac = window.contact::xxIDL_IDNumberInternal;
    $sp = Get(ScriptParameter);
    $mk = Get ( ActiveModifierKeys )
    ];"")

    /*
    Shift = 1,
    Caps Lock = 2,
    Ctrl (Windows) and Control (Mac OS) = 4,
    Alt (Windows) and Option (Mac OS) = 8,
    ⌘ (Mac OS) = 16
    */ ]
Set Field [ window::xPt_ItemIDForScript; "" ]
If [ $sp = "fieldenter" ]
    Set Variable [ $$FE; Value:1 ]
    Exit Script [  ]
End If
If [ $sp = "abbreviatenotes" ]
    Perform Script [ “UpdateLinks [ links ] ”; Parameter: window::xPt_ItemID ]
End If
If [ $sp = "fieldexit" ]
    Set Variable [ $$FE; Value:"" ]
    Exit Script [  ]
End If
If [ $sp = "cancelfuturedate" ]
    Set Variable [ $$cfd; Value:1 ]
End If
If [ $sp = "contactlinks" ]
    Set Field [ window::xPt_ContactIDShowLinkedContactLinks; Case (window::xPt_ContactIDShowLinkedContactLinks ; "" ; 1 ) ]
End If
If [ $sp = "commitrelated" ]
    If [ not IsEmpty( window.window::xxIDL_IDNumberInternal ) ]
        Set Field [ window.window::xPt_ContactID; window.window::xPt_ContactID ]
    End If
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Halt Script
End If
If [ $sp = "notestyle" ]
    Set Field [ window::xSys_NewNoteRemoveStyleFlag; Case( window::xSys_NewNoteRemoveStyleFlag = 1; ""; 1 ) ]
End If
If [ $sp = "last connect" ]
    Set Field [ window.contact::Call Last Connect; join::link.date ]
    Perform Script [ “UpdateLinks [ links ] ” ]
End If
If [ $sp = "last call" ]
    Set Field [ window.contact::Call Last Connect; Get(CurrentDate) ]
    Set Field [ window.contact::Call Next Connect Manual; Case(
        IsEmpty( window.contact::Call Next Connect Manual ) ; "" ;
        Get(CurrentDate)  ≥ window.contact::Call Next Connect Manual; "" ;
        window.contact::Call Next Connect Manual
        ) ]
End If
If [ $sp = "swapphone" ]
    If [ IsEmpty(window.contact::Phone2 Area Code) and not IsEmpty(window.contact::Phone2) ]
        Set Field [ window.contact::Phone2 Area Code; window.contact::Phone Area Code ]
    Else
        Set Variable [ $a; Value:Let ( [
            $ph[1] = window.contact::Phone2 Description;
            $ph[2] = window.contact::Phone2 Area Code ;
```

```
                $ph[3] = window.contact::Phone2
                ] ;
                "") ]
            Set Field [ window.contact::Phone2 Description; window.contact::Phone Description ]
            Set Field [ window.contact::Phone2 Area Code; window.contact::Phone Area Code ]
            Set Field [ window.contact::Phone2; window.contact::Phone ]
            Set Field [ window.contact::Phone Description; $ph[ 1 ] ]
            Set Field [ window.contact::Phone Area Code; $ph[ 2 ] ]
            Set Field [ window.contact::Phone; $ph[ 3 ] ]
        End If
End If
If [ $sp = "openweb" ]
    If [ IsEmpty(window.contact::Web Address) ]
        Set Field [ window.contact::Web Address; "www."  & Substitute(RightWords(Substitute(window.contact::
            Email;"@";"¶");1);"¶";"") ]
    Else
        Open URL [ window.contact::Web Address ]
            [ No dialog ]
    End If
End If
If [ $sp = "clear call" ]
    Set Field [ window.contact::Call Next Connect Manual; "" ]
    Set Field [ window.contact::Call Category; "" ]
    Set Field [ window.contact::Call Cycle; "" ]
    Set Field [ window.contact::Call Owner; "" ]
End If
If [ $sp = "Previous Contact" ]
    Set Field [ window::xPt_ContactID; If( Left (window::xPt_SelectedItems__LastClickedContact ; 1 ) = "c" ; window::
        xPt_SelectedItems__LastClickedContact; window::xPt_ContactID ) ]
End If
If [ $sp = "clear list" ]
    Set Field [ portal.sidebar.folder.selected::xSF_ContactType; "All Types" ]
    Set Field [ portal.sidebar.folder.selected::xSF_ContactRegion; "All Regions" ]
    Set Field [ portal.sidebar.folder.selected::xSF_ContactRole; "All Roles" ]
    Set Field [ portal.sidebar.folder.selected::xSF_ContactCallOwner; "All Call Owners" ]
End If
If [ $sp = "deletekeys" ]
    Set Variable [ $$DeleteKeyFolder; Value:window.user::xFolder_SystemDeleteKeyFolder ]
    Set Variable [ $$DeleteKeyItem; Value:window.user::xFolder_SystemDeleteKeyItem ]
    Set Variable [ $$DeleteKeyOrphaned; Value:window.user::xFolder_SystemDeleteKeyOrphaned ]
End If
If [ $sp = "folder options" ]
    If [ $$drag.note  =  "header" ]
        Set Field [ window::xID_FolderHeaderTypeExclude; Substitute (

            If( PatternCount ("¶" &  Upper( window::xID_FolderHeaderTypeExclude)  & "¶"; "¶" & $$dragheadername  & "¶" ) > 0 ;
            Substitute( "¶" &  Upper( window::xID_FolderHeaderTypeExclude) & "¶" ; "¶" & $$dragheadername  & "¶" ; "¶" ) ;
             "¶" & $$dragheadername  & "¶"  & window::xID_FolderHeaderTypeExclude  & "¶"
            )

            ; ["¶¶" ; "¶" ] ; ["¶¶" ; "¶" ] ; ["¶¶" ; "¶" ]  ) ]
    Else
        Set Field [ window::xID_FolderSelectedID; $$drag.id ]
        Perform Script [ “ItemTab” ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
    End If
End If
If [ $sp = "portal.mainlist.reminderflag" ]
```

```
        If [ not IsEmpty( portal.MainList.reminder::xSys_MainListDotChange) ]
                Set Variable [ $flag; Value:Case(
                        portal.MainList.reminder::xSys_MainListDotChange = "Red"; "2" ;
                        portal.MainList.reminder::xSys_MainListDotChange = "Blue"; "1" ;
                        portal.MainList.reminder::xSys_MainListDotChange = "Completed"; "0" ;
                        portal.MainList.reminder::xSys_MainListDotChange = "Clear"; "" ;
                         ) ]
                Set Variable [ $list; Value:If(
                        PatternCount( window::xPt_SelectedItems_MainList ; portal.MainList::xxIDL_IDNumberPortal ) > 0;  window::
                        xPt_SelectedItems_MainList ;
                        portal.MainList::xxIDL_IDNumberPortal
                        ) ]
                Set Variable [ $a; Value:Let(  [
                        $count = 0 ;
                        $LL = $list;
                        $end = WordCount( $LL )
                        ] ; ""
                        ) ]
                Loop
                        Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
                        Set Field [ window::xID_ReminderID; If(
                                Left( MiddleWords($LL;$count;1) ; 1 )   ≠   "L" ; MiddleWords($LL;$count;1) ;
                                Middle (MiddleWords($LL;$count;1); 4 ; Position ( MiddleWords($LL;$count;1) ; "LPx" ; 1; 1) - 4 )
                                 ) ]
                        Set Field [ window.reminder::Flag; $flag ]
                        Set Field [ window.reminder::Date; Case(
                                $flag = "1" ;  If( IsEmpty(window.reminder::Date); Get(CurrentDate); window.reminder::Date) ;
                                $flag = "2"; If( IsEmpty(window.reminder::Date); Get(CurrentDate); window.reminder::Date) ;
                                window.reminder::Date
                                ) ]
                        Set Field [ window.reminder::xSys_MainListDotChange; "" ]
                End Loop
                Set Field [ window::xID_ReminderID; "" ]
        End If
End If
Set Field [ window::xPt_ItemIDForScript; "" ]
Commit Records/Requests
        [ Skip data entry validation; No dialog ]
Exit Script [ Result: Let( [
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drag.PortalID = "" ;
        $$drop.ID = "" ;
        $$drop.portal = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$DraglistPortalID = "" ;
        $$Dontchange = ""
        ] ; "" ) ]
Refresh Window
        [ Flush cached join results; Flush cached external data ]
Set Field [ Case(
        $d = "Tomorrow" ; $td + 1 ;
        $d = "Next Monday" ; If( $dow =  1 ; $td + 1; $td + 2) ;
        $d = "Next Wednesday" ;  If( $dow =  1 ; $td + 3; $td + 4) ;
        $d = "2 Weeks" ; $td + 13 ;
        $d = "3 Weeks" ; $td + 20  ;
        $d = "1 Month" ; $td + 29 ;
        $d = "1.5 Month" ; $td + 45 ;
        $d = "2 Months" ; $td + 58 ;
        $d = "3 Months" ; $td + 88 ;
        $d = "6 Months" ; $td + 118 ;
        $d = "1 Year" ; $td + 360
        )
        /*
        Date
        -
        Clear
        -
        Tomorrow
        Next Monday
        Next Wednesday
        -
        2 Weeks
        3 Weeks
        -
        1 Month
        2 Months
        3 Months
        6 Months
        -
        1 Year
        */ ]
```

Open and maintenance: tools

```
Set Variable [ $a; Value:Let( [
        $ac = window.contact::xxIDL_IDNumberInternal;
        $sp = Get(ScriptParameter);
        $mk = Get ( ActiveModifierKeys )
        ];"")

        /*
        Shift = 1,
        Caps Lock = 2,
        Ctrl (Windows) and Control (Mac OS) = 4,
        Alt (Windows) and Option (Mac OS) = 8,
        ⌘ (Mac OS) = 16
        */ ]
Set Field [ window::xPt_ItemIDForScript; "" ]
If [ $sp = "aaa" ]
        Set Variable [ $$cfd; Value:1 ]
        Set Field [ window.window::xPt_ContactID; window.window::xPt_ContactID ]
End If
#
Set Variable [ $a; Value:Let(  [
        $count = 0 ;
        $LL = $list;
        $end = WordCount( $LL )
        ] ; ""
        ) ]
Loop
        Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
        Set Variable [ $link; Value: MiddleWords($LL; $count ; 1 ) ]
End Loop
#
Set Field [ window::xPt_ItemIDForScript; "" ]
Commit Records/Requests
        [ Skip data entry validation; No dialog ]
Refresh Window
        [ Flush cached join results; Flush cached external data ]
#
Exit Script [ Result: Let( [
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drag.PortalID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$DraglistPortalID = "" ;
        $$Dontchange = ""
        ] ; "" ) ]
#
Set Field [ Case(
        $d = "Tomorrow" ; $td + 1 ;
        $d = "Next Monday" ; If( $dow =  1 ; $td + 1; $td + 2) ;
        $d = "Next Wednesday" ;  If( $dow =  1 ; $td + 3; $td + 4) ;
        $d = "2 Weeks" ; $td + 13 ;
        $d = "3 Weeks" ; $td + 20  ;
        $d = "1 Month" ; $td + 29 ;
        $d = "1.5 Month" ; $td + 45 ;
        $d = "2 Months" ; $td + 58 ;
        $d = "3 Months" ; $td + 88 ;
        $d = "6 Months" ; $td + 118 ;
        $d = "1 Year" ; $td + 360
        )



        /*
        Date
```

```
    -
    Clear
    -
    Tomorrow
    Next Monday
    Next Wednesday
    -
    2 Weeks
    3 Weeks
    -
    1 Month
    2 Months
    3 Months
    6 Months
    -
    1 Year
    */ ]
#
Perform AppleScript [ Calculated AppleScript: "tell application \"ExpressTalk\"¶
        activate¶"  &

    "tell application \"System Events\" to keystroke tab¶"  &
    "tell application \"System Events\" to keystroke \"c\" using {command down}¶
    tell application \"System Events\" to keystroke \"" & $phonenumber & "\"¶" &

    "tell application \"System Events\" to keystroke tab¶"  &
    "tell application \"System Events\" to keystroke \"c\" using {command down}¶
    tell application \"System Events\" to keystroke \"" & $phonenumber & "\"¶" &

    "tell application \"System Events\" to keystroke tab¶"  &

    "tell application \"System Events\" to keystroke \"1\" using {command down}¶"  &

    "end tell" ]
#
Perform AppleScript [ Native AppleScript: do shell script "sudo renice -18 -p `ps -ax | grep '[/]FileMaker' | awk '{print $1}'`" ]
#EDITOR=pico sudo visudo
%staff ALL=NOPASSWD: /usr/bin/renice
#
Set Variable [ $a; Value:GetLayoutObjectAttribute("tab.note.2";"isFrontTabPanel")  =  "1" ]
#
Loop
    Exit Loop If [ Case(
        $count < 1 ; Let( [
                $count = 1 ;
                $end = PatternCount( $ll ; "¶" )
        ] ; "" ) ;
        $count  ≥ $end or $end < 1;           Let( $count = 0 ; 1) ;
        Let( $count = $count + 1 ; "" )
        ) ]
    Set Variable [ $link; Value: MiddleWords($LL; $count ; 1 ) ]
End Loop
#
#Code:

#!/bin/sh
USERID=`id -u`;
if [[ $USERID == 501 ]]; then
        /System/Library/CoreServices/Menu\ Extras/User.menu/Contents/Resources/CGSession -switchToUserID 502
else
        /System/Library/CoreServices/Menu\ Extras/User.menu/Contents/Resources/CGSession -switchToUserID 501
fi;
```

Open and maintenance: MultipleSelection [single, required, update, notechoice, checkbox]

```
Set Variable [ $rs; Value:Get ( RecordOpenState ) ]
#single=just one
required=must have one
update=update join record
notechoice=for note entry
If [ Get(ScriptParameter) = "notechoice" ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Go to Object [ Object Name: "field.note.add" ]
Else If [ PatternCount( Get(ScriptParameter) ; "checkbox" ) > 0 ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
Else If [ PatternCount( Get(ScriptParameter) ; "notechoice" ) > 0 ]
Else If [ $rs = 0 ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
Else
    Set Variable [ $any_field; Value:Get ( ActiveFieldTableName ) & "::" & Get ( ActiveFieldName ) ]
    Set Variable [ $selected; Value: Evaluate ( $any_field ) ]
    Undo/Redo
        [ Undo ]
    Set Variable [ $before; Value: Evaluate ( $any_field ) ]
    Set Field By Name [ $any_field; Let ( a  = Case(
        $selected = "(hold ⌘ for multiple)" ; $before ;
        PatternCount( Get(ScriptParameter) ; "required" ) > 0  and $before = $selected ; $selected ;
        PatternCount( Get(ScriptParameter) ; "required" ) > 0  and IsEmpty( $selected ) ; $before ;
        PatternCount( Get(ScriptParameter) ; "required" )  = 0  and $before = $selected; "" ;
        PatternCount( Get(ScriptParameter) ; "single" ) > 0 ; $selected ;
        Get ( ActiveModifierKeys )  ≠ 16; $selected ;
        Length( FilterValues(  $before ; $selected  ) ) > 0 ;
        Substitute( "#S#"  & $before & "#S#"  ; [ "¶" ; "#S#¶#S#"  ] ; [ "#S#"  & $selected  & "#S#¶" ; "" ] ; [ "#S#"  & $selected  & "#S#" ;
        "" ] ; ["#S#" ; "" ] ) ;
        Case( LeftValues( $selected  & "¶" ; 1 )  < LeftValues( $before  & "¶"; 1 ) ; $selected  & "¶" & $before ; $before  & "¶" &
        $selected )
        )
        ;
        Substitute ( Trim (Substitute( a ; [ " " ; "#S#S#" ] ; [ "¶" ; " " ] ) ) ; [ " " ; "¶" ] ; [ "#S#S#" ; " " ] )
        )

        // there is a bug in FM, the lowest alphabetical value in the value list must be first in the field or it won't show as checked ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
End If
If [ PatternCount( Get(ScriptParameter) ; "update" ) > 0 ]
    Perform Script [ “UpdateLinks [ links ] ” ]
End If
Set Field [ window::xPt_ItemIDForScript; "" ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Exit Script [  ]
Show Custom Dialog [ Message: Get ( RecordOpenState ); Buttons: “OK”, “Cancel” ]
#
#
```

Open and maintenance: Timer

```
If [ PatternCount(Get ( LayoutName ) ; "Main" ) > 0 and window::xSys_PhoneHangup  ≠  1 ]
    Freeze Window
    Set Field [ window::xSys_TimeZoneCurrent; "" ]
    Set Field [ window::xSys_TimeZoneCurrent; window::xSys_TimeZoneCurrent &
        Let( [
        t = Get(CurrentTime) ; tz = "-1"; mh = Hour(t) + GetAsNumber(tz) ; m = Minute(t); mmh = (mh *60) + m;
        p = Case(
        mmh < window.user::xPrefSys_TZhoursCalc[1]  ; "";
        mmh < window.user::xPrefSys_TZhoursCalc[2] ; tz;
        mmh < window.user::xPrefSys_TZhoursCalc[3] ; "" ;
        mmh < window.user::xPrefSys_TZhoursCalc[4] ; tz;
        ""
        )
        ];
        Case(
        IsEmpty(p) ; "";
        IsEmpty(window::xSys_TimeZoneCurrent) ; p ;
         "¶" & p
        )
        ) ]
    Set Field [ window::xSys_TimeZoneCurrent; window::xSys_TimeZoneCurrent &
        Let( [
        t = Get(CurrentTime) ; tz = "+0"; mh = Hour(t) + GetAsNumber(tz) ; m = Minute(t); mmh = (mh *60) + m;
        p = Case(
        mmh < window.user::xPrefSys_TZhoursCalc[1]  ; "";
        mmh < window.user::xPrefSys_TZhoursCalc[2] ; tz;
        mmh < window.user::xPrefSys_TZhoursCalc[3] ; "" ;
        mmh < window.user::xPrefSys_TZhoursCalc[4] ; tz;
        ""
        )
        ];
        Case(
        IsEmpty(p) ; "";
        IsEmpty(window::xSys_TimeZoneCurrent) ; p ;
         "¶" & p
        )
        ) ]
    Set Field [ window::xSys_TimeZoneCurrent; window::xSys_TimeZoneCurrent &
        Let( [
        t = Get(CurrentTime) ; tz = "+1"; mh = Hour(t) + GetAsNumber(tz) ; m = Minute(t); mmh = (mh *60) + m;
        p = Case(
        mmh < window.user::xPrefSys_TZhoursCalc[1]  ; "";
        mmh < window.user::xPrefSys_TZhoursCalc[2] ; tz;
        mmh < window.user::xPrefSys_TZhoursCalc[3] ; "" ;
        mmh < window.user::xPrefSys_TZhoursCalc[4] ; tz;
        ""
        )
        ];
        Case(
        IsEmpty(p) ; "";
        IsEmpty(window::xSys_TimeZoneCurrent) ; p ;
         "¶" & p
        )
        ) ]
    Set Field [ window::xSys_TimeZoneCurrent; window::xSys_TimeZoneCurrent &
        Let( [
        t = Get(CurrentTime) ; tz = "+2"; mh = Hour(t) + GetAsNumber(tz) ; m = Minute(t); mmh = (mh *60) + m;
        p = Case(
        mmh < window.user::xPrefSys_TZhoursCalc[1]  ; "";
        mmh < window.user::xPrefSys_TZhoursCalc[2] ; tz;
        mmh < window.user::xPrefSys_TZhoursCalc[3] ; "" ;
        mmh < window.user::xPrefSys_TZhoursCalc[4] ; tz;
        ""
        )
        ];
        Case(
        IsEmpty(p) ; "";
        IsEmpty(window::xSys_TimeZoneCurrent) ; p ;
         "¶" & p
        )
        ) ]
    Set Field [ window::xSys_TimeZoneCurrent; window::xSys_TimeZoneCurrent &
        Let( [
        t = Get(CurrentTime) ; tz = "+3"; mh = Hour(t) + GetAsNumber(tz) ; m = Minute(t); mmh = (mh *60) + m;
        p = Case(
        mmh < window.user::xPrefSys_TZhoursCalc[1]  ; "";
        mmh < window.user::xPrefSys_TZhoursCalc[2] ; tz;
        mmh < window.user::xPrefSys_TZhoursCalc[3] ; "" ;
        mmh < window.user::xPrefSys_TZhoursCalc[4] ; tz;
        ""
        )
        ];
        Case(
        IsEmpty(p) ; "";
        IsEmpty(window::xSys_TimeZoneCurrent) ; p ;
         "¶" & p
        )
        ) ]
    Set Field [ window::xPt_ItemIDForScript; "" ]
    If [ IsEmpty(Get ( ActiveFieldName )) ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
        If [ $$systemdate  ≠ Get(CurrentDate) ]
            Perform Script [ “Daily Reminders” ]
        End If
    End If
    Perform Script [ “Export Todo List” ]
End If
If [ Case(
    Get ( ActiveModifierKeys ) = 1 or Get ( ActiveModifierKeys ) = 16 ; "" ;
    Get(CurrentTime) > window.user::xPrefUser_ShutdownTime and
    Get(CurrentTime) < window.user::xPrefUser_ShutdownTime + 60 ; 1 ;
    ""
    ) ]
    If [ 1 = 2 ]
        Set Variable [ $$as; Value:"tell application \"FileMaker Pro\" to set cellValue of repetition 1 of cell \"clipboard\" of table \"system\"
        of database \"dddddd\" to \"\"" & "¶" &
            "set someVariable to \"\"" & "¶" &

            "try¶" &

            "set someVariable to do shell script \"cccccc\" user name \"aaaaaa\" password \"pppppp\" with administrator privileges¶" &

            "on error errorMsg¶" &
               "set someVariable to \"error \"  & errorMsg ¶" &

            "end try¶" &

            "if someVariable is \"error\" then¶" &
            "else¶" &
            "end if¶" &

            "tell application \"FileMaker Pro\" to set cellValue of repetition 1 of cell \"clipboard\" of table \"system\"  of database
            \"dddddd\" to someVariable¶"

            //"display dialog someVariable¶"
             ]
        Set Variable [ $cmd; Value:"/sbin/shutdown -h +2 ; /usr/bin/osascript -e 'tell app \"Finder\" to display alert \"Shutting down in two
            minutes.  All apps will be shutdown without saving\" as warning giving up after 5'" ]
        Perform AppleScript [ Calculated AppleScript: /*Substitute( $$as ; [ "cccccc" ; $cmd ] ; ["aaaaaa" ; hosts.all::_OSadmin ] ;
            [ "pppppp" ; $$pw ] ; [ "dddddd" ; Get ( FileName ) ] )*/ ]
        If [ /*LeftWords( System::clipboard ; 1 )= "error"*/ ]
            Commit Records/Requests
                [ Skip data entry validation; No dialog ]
        End If
    End If
    Exit Application
End If
Exit Script [  ]
```

Open and maintenance: Layout key trigger

```
If [ $$FE ]
    Exit Script [  ]
End If
If [ IsEmpty(Get ( ActiveFieldName ) ) ]
    Set Variable [ $a; Value:Let( [
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$Dontchange = ""
        ] ; "" ) ]
    #START If not in a field
    If [ Code (Get ( TriggerKeystroke ) ) = 13 or // return
        Code (Get ( TriggerKeystroke ) ) = 32     // space ]
        #Space
        If [ WordCount (Evaluate (  "xPt_SelectedItems_" & window::xPt_SelectedItems__CurrentClickedPortal)   ) = 1 ]
            Set Variable [ $$EnterNote; Value:"key" ]
            Perform Script [ “DoubleClicked [ link (optional), $$EnterNote , $$drag.portal]” ]
            Set Variable [ $$EnterNote; Value:"" ]
            Commit Records/Requests
                [ Skip data entry validation; No dialog ]
            Perform Script [ “ItemTab” ]
        End If
        #Left or Right Arrow
    Else If [
        Code (Get ( TriggerKeystroke ) ) = 30     // right arrow ]
        Go to Object [ Object Name: "tab.note.2" ]
    Else If [ Code (Get ( TriggerKeystroke ) ) = 28  // left arrow
     ]
        If [ GetLayoutObjectAttribute("tab.note.2";"isFrontTabPanel")  =  "1" ]
            Perform Script [ “NoteButtons [ enter, exit, edit, connectdate ]” ]
        End If
    Else If [ Code (Get ( TriggerKeystroke ) ) = 31 // down arrow ]
        #Down Arrow
        Set Field [ window::xPt_SelectedItems_MainList; Let( [
            $O = window::xPt_SelectedItems_MainList ;
            $L =List( portal.MainList::xxIDL_IDNumberPortal ) ;
            $Lcount = WordCount ($L) ;
            $FO = FilterValues( $L ; $O)  ;
            $PRI = Case(
            IsEmpty( window::xPt_SelectedItems__CurrentClickedPortalRowMainlist ) ; 1;
            window::xPt_SelectedItems__CurrentClickedPortalRowMainlist < 1; 1;
            window::xPt_SelectedItems__CurrentClickedPortalRowMainlist > $lcount; $lcount ;
            window::xPt_SelectedItems__CurrentClickedPortalRowMainlist  )
            ] ;
            Case(
            WordCount( $FO ) > 0 ; $FO ;
            MiddleWords( $L ; $PRI ; 1 )
            )
            ) ]
        If [ 1 = 2 ]
            Set Variable [ $r; Value:Let( [
                $portalID = LeftWords( window::xPt_SelectedItems_MainList ; 1 ) ;
                $PRI = Position (  Filter( Substitute(  Substitute( "TTT" &  $L  & "TTT"; "¶" ; "TTT¶TTT" ) ; "TTT"  & $PortalID  & "TTT" ;
                "#") ; "#¶"  ) ; "#" ;1; 1)  ;
                $PR = Case( $PRI < 1 ; 1;
                $PRI  ≥ $Lcount; $lcount ;
                $PRI + 1 ) ;
                $$drag.portalID = MiddleWords( $L ; $PR ; 1 ) ;
                $$drag.portal ="mainlist"
                ]; ""
                ) ]
        End If
        Set Variable [ $r; Value:Let( [
            $portalID = LeftWords( window::xPt_SelectedItems_MainList ; 1 ) ;
            $PRI = WordCount( Replace ( $l  ; Position ( $l ; $PortalID ; 1 ; 1 ) ; 950000 ; "" ) ) + 1  ;
            $PR = Case( $PRI < 1 ; 1;
            $PRI  ≥ $Lcount; $lcount ;
            $PRI + 1 ) ;
            $$drag.portalID = MiddleWords( $L ; $PR ; 1 ) ;
            $$drag.portal ="mainlist"
            ]; ""
            ) ]
        Set Field [ window::xPt_SelectedItems__CurrentClickedPortalRowMainlist; $PRI ]
        If [ window::xPt_MainListLayout = "importemail" ]
            Perform Script [ “ImportEmailWindow [ link ]”; Parameter: $$drag.portalID ]
        Else
            Perform Script [ “QuickClicked ” ]
        End If
        Set Variable [ $$ClickTimer; Value:Get(CurrentTime) - 3 ]
        Go to Object [ Object Name: "portal.mainlist" ]
        Go to Portal Row [ $PR -  2 ]
            [ Select; No dialog ]
        Go to Portal Row [ $PR + 2 ]
            [ Select; No dialog ]
        Go to Portal Row [ $PR ]
            [ Select; No dialog ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
    Else If [ Code (Get ( TriggerKeystroke ) ) = 29  // up arrow ]
        Set Field [ window::xPt_SelectedItems_MainList; Let( [
            $O = window::xPt_SelectedItems_MainList ;
            $L =List( portal.MainList::xxIDL_IDNumberPortal ) ;
            //$Lcount = WordCount ($L) ;
            $FO = FilterValues( $L ; $O)  ;
            $PRI = Case(
            IsEmpty( portal.sidebar.folder.selected::xFSelected_Selected_CurrentPortalRow ) ; 1;
            portal.sidebar.folder.selected::xFSelected_Selected_CurrentPortalRow < 1; 1;
            portal.sidebar.folder.selected::xFSelected_Selected_CurrentPortalRow > $lcount; $lcount ;
            portal.sidebar.folder.selected::xFSelected_Selected_CurrentPortalRow  )
            ] ;
            Case(
            WordCount( $FO ) > 0 ; $FO ;
            MiddleWords( $L ; $PRI ; 1 )
            )
            ) ]
        If [ 1 = 2 ]
            Set Variable [ $r; Value:Let( [
                $portalID = LeftWords( window::xPt_SelectedItems_MainList ; 1 ) ;
                $PRI = Position (  Filter( Substitute(  Substitute( "TTT" &  $L  & "TTT"; "¶" ; "TTT¶TTT" ) ; "TTT"  & $PortalID  & "TTT" ;
                "#") ; "#¶"  ) ; "#" ;1; 1)  ;
                $PR = Case( $PRI < 2; 1; $PRI - 1 ) ;
                $$drag.portalID = MiddleWords( $L ; $PR ; 1 ) ;
                $$drag.portal ="mainlist"
                ]; ""
                ) ]
        End If
        Set Variable [ $r; Value:Let( [
            $portalID = LeftWords( window::xPt_SelectedItems_MainList ; 1 ) ;
            $PRI = WordCount( Replace ( $l  ; Position ( $l ; $PortalID ; 1 ; 1 ) ; 950000 ; "" ) ) + 1  ;
            $PR = Case( $PRI < 2; 1; $PRI - 1 ) ;
            $$drag.portalID = MiddleWords( $L ; $PR ; 1 ) ;
            $$drag.portal ="mainlist"
            ]; ""
            ) ]
        If [ window::xPt_MainListLayout = "importemail" ]
            Perform Script [ “ImportEmailWindow [ link ]”; Parameter: $$drag.portalID ]
        Else
            Perform Script [ “QuickClicked ” ]
```

```
End If
    Set Variable [ $$ClickTimer; Value:Get(CurrentTime) - 3 ]
    Go to Object [ Object Name: "portal.mainlist" ]
    Go to Portal Row [ $PR -  2 ]
        [ Select; No dialog ]
    Go to Portal Row [ $PR + 2 ]
        [ Select; No dialog ]
    Go to Portal Row [ $PR ]
        [ Select; No dialog ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
Else If [ Code (Get ( TriggerKeystroke ) ) = 8  or // delete key
    Code (Get ( TriggerKeystroke ) ) = 127 ]
    #Delete
    If [ Case(
        window::xID_FolderSelectedID = $$TrashID and window::xPt_SelectedItems__CurrentClickedPortal = "MainList"; "" ;
        window::xPt_SelectedItems__CurrentClickedPortal = "LinkToItemList"; "" ;
        1) ]
        Set Field [ window::xPt_SelectedItems__CurrentClickedPortalSelectedIDs; Let( a = FilterValues (
            Evaluate ( "xPt_SelectedItems_" & window::xPt_SelectedItems__CurrentClickedPortal)  & "¶" ;
            List( Evaluate( "portal." & window::xPt_SelectedItems__CurrentClickedPortal & "::xxIDL_IDNumberPortal" )   )
             )

            ;
            Substitute( Trim( Substitute ( a ; "¶" ; " "  ) ) ; " " ; "¶" )
            ) ]
        Set Variable [ $$Drag.portal; Value:window::xPt_SelectedItems__CurrentClickedPortal ]
        Set Variable [ $$DragList; Value:window::xPt_SelectedItems__CurrentClickedPortalSelectedIDs ]
        If [ not IsEmpty($$draglist ) ]
            Set Variable [ $mixed; Value:Let( [
                $listclean = List( Evaluate( "portal." & window::xPt_SelectedItems__CurrentClickedPortal & "::
                xxIDL_IDNumberInternal" )  ) ;
                $sl = window::xPt_SelectedItems__CurrentClickedPortalSelectedIDs
                ] ;
                Case (
                WordCount( FilterValues( $sl ; $listclean)  ) < 1;  "LCx"  ;
                PatternCount( $sl ; "LCx" ) > 0 ; "Mixed" ;
                "Clean"
                )
                ) ]
            If [ $mixed =  "Mixed" ]
            Else
                #Move To Removed Folder
                Move To Trash
                Remove, Clear Reminder
                Remove

                If [ $$drag.portal = "ItemLinkList" ]
                    Set Variable [ $$drop.id; Value:window::xPt_ItemID ]
                    Perform Script [ “LinkRemove [$$drop.id, $$draglist, cleans]” ]
                Else If [ $$drag.portal = "ContactLinkList" ]
                    Set Variable [ $$drop.id; Value:window::xPt_ContactID ]
                    Perform Script [ “LinkRemove [$$drop.id, $$draglist, cleans]” ]
                Else If [ $$drag.portal = "Notes" ]
                    Set Variable [ $$drop.id; Value:window::xPt_ContactID ]
                    Perform Script [ “LinkRemove [$$drop.id, $$draglist, cleans]” ]
                Else If [ $$drag.portal = "CompanyList" ]
                    Set Variable [ $$drop.id; Value:window::xPt_ContactID ]
                    Perform Script [ “LinkRemove [$$drop.id, $$draglist, cleans]” ]
                Else If [ $$drag.portal = "MainList" ]
                    If [ window::xPt_MainListLayout  =  "importemail" ]
                        Show Custom Dialog [ Title: "Delete"; Message: Case(
                            WordCount( $$draglist ) = 1 ; "Permanently delete this email?";
                            "Permanently delete these emails?"
                            ); Buttons: “Cancel”, “Ok” ]
                            If [ Get ( LastMessageChoice ) = 1 ]
                                Set Variable [ $a; Value:Let( [
                                    $$drop.note = "" ;
                                    $$drag.ID = "" ;
                                    $$drag.PortalID = "" ;
                                    $$drop.ID = "" ;
                                    $$drag.portal = "";
                                    $$drag.note = "";
                                    $$dragheadername = "";
                                    $$drop.sidebarID = "" ;
                                    $$drag.sidebarID = ""  ;
                                    $$Drag.foldertype = "" ;
                                    $$Draglist = "" ;
                                    $$DraglistPortalID = "" ;
                                    $$Dontchange = ""
                                    ] ; "" ) ]
                                Set Field [ window::xPt_ItemIDForScript; "" ]
                                Halt Script
                            End If
                            Perform Script [ “Move/Copy/ClearToAnyFolder [ $$drop.ID [1], $$DragList [2] cleans, g(sp) = from
                                folders ] ”; Parameter: Let( $$Drop.Id =$$TrashID ;  "") ]
                        Else If [ $mixed =  "Clean" ]
                            Set Variable [ $$drop.id; Value:Case(
                                $$DeleteKeyFolder = "Move To Trash" ; $$TrashID;
                                ""
                                ) ]
                            Perform Script [ “Move/Copy/ClearToAnyFolder [ $$drop.ID [1], $$DragList [2] cleans, g(sp) = from
                                folders ] ”; Parameter: window::xID_FolderSelectedID ]
                        Else
                            Set Variable [ $$drop.id; Value:"" ]
                            Perform Script [ “LinkRemove [$$drop.id, $$draglist, cleans]” ]
                        End If
                    Else If [ $$drag.portal = "LinkToItemList" ]
                        Set Variable [ $a; Value:Let( [
                            $$drop.note = "" ;
                            $$drag.ID = "" ;
                            $$drag.PortalID = "" ;
                            $$drop.ID = "" ;
                            $$drag.portal = "";
                            $$drag.note = "";
                            $$dragheadername = "";
                            $$drop.sidebarID = "" ;
                            $$drag.sidebarID = ""  ;
                            $$Drag.foldertype = "" ;
                            $$Draglist = "" ;
                            $$DraglistPortalID = "" ;
                            $$Dontchange = ""
                            ] ; "" ) ]
                        Set Field [ window::xPt_ItemIDForScript; "" ]
                        Halt Script
                    End If
                    Set Field [ window::xPt_SelectedItems__CurrentClickedPortal; "" ]
                    Set Field [ window::xPt_SelectedItems__CurrentClickedPortalID; "" ]
                    Set Field [ window::xPt_SelectedItems__LastClickedPortalID; "" ]
                    Set Field [ window::xPt_SelectedItems__LastClickedPortal; "" ]
                End If
            End If
        End If
    Else If [ Code (Get ( TriggerKeystroke ) ) = 1  and Get ( TriggerModifierKeys ) = 4 // crt-A ]
        #crtl-A
        Set Field By Name [ window::xPt_SelectedItems__CurrentClickedPortalFN; List( Evaluate( "portal." & window::
            xPt_SelectedItems__CurrentClickedPortal & "::xxIDL_IDNumberInternal" )  ) ]
    Else If [ Code (Get ( TriggerKeystroke ) ) = 6 and Get ( TriggerModifierKeys ) = 4 ]
        #crt-F
        If [ PatternCount( LayoutObjectNames ( "customers"  ; Get(LayoutName) ) ; "portal.mainlist" ) > 0 ]
```

```
            Commit Records/Requests
                [ Skip data entry validation; No dialog ]
            Perform Script [ “SpotlightEntry [enter, close]”; Parameter: "enter" ]
        End If
        #esc
    Else If [ Code (Get ( TriggerKeystroke ) ) = 27 ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
        #tab
    Else If [ Code (Get ( TriggerKeystroke ) ) = 9  and Get ( TriggerModifierKeys ) = 1 // shift tab ]
        Go to Object [ Object Name: "NewNote" ]
    Else If [ Code (Get ( TriggerKeystroke ) ) = 9  // tab ]
        Go to Object [  ]
        Exit Script [  ]
    Else If [ 1 = 2

        //Code (Get ( TriggerKeystroke ) ) = 9  // tab ]
    Else If [ 1 = 2

        //Code (Get ( TriggerKeystroke ) ) = 9  // tab ]
        If [ Get ( ActiveLayoutObjectName ) ="button.add" and window::<Field Missing> = "communication" ]
            Set Field [ window::<Field Missing>; "Note" ]
        Else If [ Get ( ActiveLayoutObjectName ) ="button.differentvm" and  window::<Field Missing> = "note" ]
            Set Field [ window::<Field Missing>; "Communication" ]
            Go to Object [ Object Name: "button.add" ]
        Else
            Go to Object [ Object Name: Case(
                Get ( ActiveLayoutObjectName ) ="button.add" ; "button.notin" ;
                Get ( ActiveLayoutObjectName ) ="button.notin" ; "button.leftvm" ;
                Get ( ActiveLayoutObjectName ) ="button.leftvm" ; "button.wrongnumber" ;
                Get ( ActiveLayoutObjectName ) ="button.wrongnumber" ; "button.differentvm" ;
                Get ( ActiveLayoutObjectName ) ="button.differentvm" ; "button.add"  ;
                "button.add" ) ]
        End If
    End If
    Set Variable [ $a; Value:Let( [
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drag.PortalID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$DraglistPortalID = "" ;
        $$Dontchange = ""
        ] ; "" ) ]
    Set Field [ window::xPt_ItemIDForScript; "" ]
    Halt Script
    Show Custom Dialog [ Message: Code( Get ( TriggerKeystroke ) ) &  "¶" &  Get ( TriggerModifierKeys ); Buttons: “OK”, “Cancel” ]
    Exit Script [  ]
    #END Not in a Field
End If
Exit Script [  ]
#
#
If [  ]
Else
    Exit Script [  ]
    If [ Code (Get ( TriggerKeystroke ) ) = 27 ]
```

```
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
    Else If [ 1 = 2
        //Code (Get ( TriggerKeystroke ) ) = 9 ]
        If [ Get( ActiveFieldName) = "xNew_Note" ]
            Go to Object [ Object Name: "button.add" ]
        End If
    Else If [ Get( ActiveFieldName) = "xSys_ContactNotes" ]
        If [ Code (Get ( TriggerKeystroke ) ) = 28  // left arrow
            and
            GetLayoutObjectAttribute("tab.note.2";"isFrontTabPanel")  =  "1" ]
            Perform Script [ “NoteButtons [ enter, exit, edit, connectdate ]” ]
        End If
    Else
        Exit Script [  ]
    End If
    Halt Script
End If
Exit Script [  ]
#
#
#/*
Shift = 1,
Caps Lock = 2,
Ctrl (Windows) and Control (Mac OS) = 4,
Alt (Windows) and Option (Mac OS) = 8,
⌘ (Mac OS) = 16
note: linefeed is 32

*/
Show Custom Dialog [ Message: Code( Get ( TriggerKeystroke ) ) &  "¶" &  Get ( TriggerModifierKeys ); Buttons: “OK”, “Cancel” ]
Exit Script [ Result: // $link
    $link  &
    Let( [
    $$drop.note = "" ;
    $$drag.ID = "" ;
    $$drag.PortalID = "" ;
    $$drop.ID = "" ;
    $$drag.portal = "";
    $$drag.note = "";
    $$dragheadername = "";
    $$drop.sidebarID = "" ;
    $$drag.sidebarID = ""  ;
    $$Drag.foldertype = "" ;
    $$Draglist = "" ;
    $$DraglistPortalID = "" ;
    $$Dontchange = ""
    ] ; "" ) ]
Exit Script [  ]
Show Custom Dialog [ Message: Code( Get ( TriggerKeystroke ) ) &  "¶" &  Get ( TriggerModifierKeys ); Buttons: “OK”, “Cancel” ]
Perform Script [ “Move/Copy/ClearToAnyFolder [ $$drop.ID [1], $$DragList [2] cleans, g(sp) = from folders ] ”; Parameter: Let( $$drop.id
    = $$removedID; "") ]
Go to Portal Row [ $PR ]
    [ Select; No dialog ]
Show Custom Dialog [ Message: "sml " & window::xPt_SelectedItems_MainList &
    "¶pri " & $PRI &
    "¶por slf f" & portal.sidebar.folder.selected::xFSelected_Selected_CurrentPortalRow
     & "¶cur p[ort " & Get ( ActivePortalRowNumber )
     & "¶ win:pr " & window::xPt_SelectedItems__CurrentClickedPortalRowMainlist; Buttons: “OK”, “Cancel” ]
Show Custom Dialog [ Message:  "¶window::xID_FolderSelectedID" & window::xID_FolderSelectedID
     & "¶$$RemovedID" & $$RemovedID
     & "¶window::xPt_SelectedItems__CurrentClickedPortal" & window::xPt_SelectedItems__CurrentClickedPortal
     & "¶window::xID_FolderSelectedID" & window::xID_FolderSelectedID
     & "¶$$TrashID" & $$TrashID
     & "¶window::xPt_SelectedItems__CurrentClickedPortal" &  window::xPt_SelectedItems__CurrentClickedPortal
     & "¶window::xPt_SelectedItems__CurrentClickedPortal" & window::xPt_SelectedItems__CurrentClickedPortal; Buttons: “OK”,
    “Cancel” ]
```

Open and maintenance: Import Contacts Autoenter, create notes 1, 2, 3

```
Freeze Window
#1 = note field, 2 = create note, 3 = folder name, 4, 5 = call category, 6= call back owner
Set Field [ window::xSys_ImportOptions; "Imported "  & Get(CurrentDate) ]
Set Field [ window::xSys_ImportOptions[2]; "Imported "  & Get(CurrentDate) ]
Set Field [ window::xSys_ImportOptions[3]; "Imported "  & Get(CurrentDate) ]
Set Field [ window::xSys_ImportOptions[4]; "" ]
Set Field [ window::xSys_ImportOptions[5]; "" ]
Set Field [ window::xSys_ImportOptions[6]; "" ]
Set Field [ window::xSys_ImportOptions[7]; "" ]
Show Custom Dialog [ Title: "Import Contacts"; Buttons: “OK”, “Cancel”; Input #1: window::xSys_ImportOptions, "Import Note Field"; Input
    #2: window::xSys_ImportOptions[2], "Create note in notepad"; Input #3: window::xSys_ImportOptions[3], "Folder Name" ]
If [ Get ( LastMessageChoice ) = 2 ]
    Exit Script [  ]
End If
Set Field [ window::sys.GeneralDialogAnswers; "" ]
Show Custom Dialog [ Title: "Import Contacs 2"; Message: Substitute( ValueListItems ( Get(FileName) ; "Contact.call category" ) ; ["¶" ; ",
" ] ; ["-" ; "" ] ) & "¶¶" &
    "If cleaning up multiple contacts: import names -> name, phones -> phone (no area code), emails -> email" & "¶" &
    "The left 2 words of name will be taken as the contact name"; Buttons: “OK”, “Cancel”; Input #1: window::xSys_ImportOptions[5], "Call
    Back Category"; Input #2: window::xSys_ImportOptions[6], "Call Back Owner"; Input #3: window::sys.GeneralDialogAnswers, "Clean
    up multipe Contacts/phone/email in the same record? (Y,N)" ]
If [ Get ( LastMessageChoice ) = 2 ]
    Exit Script [  ]
End If
New Window [ Name: "~temp~"; Height: 80; Width: 80; Top: -400 ]
Go to Layout [ “data.contact” (item.contact) ]
#Import and replace specific field contents
Import Records [ Target: “item.contact”; Method: Add; Character Set: “Mac Roman”; Field Mapping: Source field 1 import to item.contact::
Status
    Source field 3 import to item.contact::Company Name
    Source field 4 import to item.contact::Name
    Source field 9 import to item.contact::Lead Number
    Source field 10 import to item.contact::Ship Address
    Source field 11 import to item.contact::Ship City
    Source field 12 import to item.contact::Ship State
    Source field 13 import to item.contact::Ship Zip
    Source field 14 import to item.contact::Import Note 2
    Source field 15 import to item.contact::Import Note 3
    Source field 16 import to item.contact::Call Next Connect Manual
    Source field 17 import to item.contact::Call Referral
    Source field 18 import to item.contact::Call Referral to
    Source field 22 import to item.contact::Note Import
    Source field 23 import to item.contact::Import Note 1
    Source field 24 import to item.contact::Department
    Source field 25 import to item.contact::Email
    Source field 26 import to item.contact::Fax AreaCode
    Source field 27 import to item.contact::Fax Number
    Source field 30 import to item.contact::Image
    Source field 34 import to item.contact::Name Note
    Source field 36 import to item.contact::Phone
    Source field 37 import to item.contact::Phone Area Code
    Source field 39 import to item.contact::Phone2
    Source field 40 import to item.contact::Phone2 Area Code
    Source field 41 import to item.contact::Phone2 Description
    Source field 43 import to item.contact::Rank
    Source field 44 import to item.contact::Region
    Source field 45 import to item.contact::Region Overide
    Source field 47 import to item.contact::Role
    Source field 48 import to item.contact::Call Last Attempt
    Source field 51 import to item.contact::Ship Attention
    Source field 52 import to item.contact::Call Last Connect
    Source field 53 import to item.contact::Call Last email
    Source field 54 import to item.contact::Call Last VM
    Source field 55 import to item.contact::Account Number
    Source field 56 import to item.contact::Title
    Source field 57 import to item.contact::Account Type
    Source field 58 import to item.contact::User Field
    Source field 59 import to item.contact::User Field Date
    Source field 60 import to item.contact::Web Address
    Source field 66 import to item.contact::xPrint_FaxPages
    Source field 67 import to item.contact::xPrint_FaxText
    Source field 69 import to item.contact::xPrint_LabelField
    Source field 70 import to item.contact::xPrint_LabelOn
    Source field 71 import to item.contact::xPrint_LabelSortNumber
    Source field 72 import to item.contact::xPrint_LabelStartNumber
    Source field 77 import to item.contact::xSys_ChangableFieldChoice
    Source field 79 import to item.contact::xSys_DateCreated
    Source field 82 import to item.contact::xSys_LeadNumber
    Source field 84 import to item.contact::xSys_PrintKeyChoice
    Source field 86 import to item.contact::xSys_TimeCreated
    Source field 89 import to item.contact::xxCF_Modified
    Source field 91 import to item.contact::xxIDL_IDNumberInternal
    Source field 94 import to item.contact::xxIDL_Owner
    Source field 95 import to item.contact::xxIDL_Trash
    Source field 98 import to item.contact::Call Cycle
    Source field 99 import to item.contact::Phone Description
    Source field 100 import to item.contact::xSys_DBN1to10
    Source field 103 import to item.contact::Status Reason
    Source field 105 import to item.contact::Call Owner
    Source field 108 import to item.contact::Call Category Color
    Source field 111 import to item.contact::Company Category
    Source field 113 import to item.contact::xxtransfer
    Source field 114 import to item.contact::Import ID number ]
Replace Field Contents [ item.contact::xxIDL_IDNumberInternal; Replace with calculation: If( IsEmpty( item.contact::
xxIDL_IDNumberInternal ) ;
    Let( $a =
    "C" &
    Right("00" & Day(Get(CurrentDate));2)  &
    Right("00" & Month(Get(CurrentDate));2) &
    Right(Year(Get(CurrentDate));2)  &
    Right("00000" & GetAsText(Get(RecordID));5) &
    Right("0000" & GetAsText(Int(9999*Random));4)
    ; $a )  & Let( $acimport = $acimport  & "¶" & $a ; "" )
    ;
     item.contact::xxIDL_IDNumberInternal & Let($acimport = $acimport  & "¶"  & item.contact::xxIDL_IDNumberInternal; "")
    ) ]
    [ No dialog ]
Replace Field Contents [ item.contact::Note Import; Replace with calculation: window::xSys_ImportOptions[1] ]
    [ No dialog ]
Replace Field Contents [ item.contact::Call Category; Replace with calculation: window::xSys_ImportOptions[5] ]
    [ No dialog ]
Replace Field Contents [ item.contact::Call Owner; Replace with calculation: window::xSys_ImportOptions[6] ]
    [ No dialog ]
Close Window [ Current Window ]
#
Set Variable [ $acimport; Value: Substitute( Trim( Substitute( $acimport ; [" "; "@" ] ; ["¶" ; " " ] ) ) ; [" "; "¶" ] ; [ "@" ; " "] ; ["¶¶"; "¶" ]) ]
If [ WordCount($acimport) = 0 ]
    Exit Script [ Result: Let( [
        $acimport ="" ;
        $$NewFolder[1] = "" ;  //type
        $$NewFolder[2] = "" ;  //name
        $$NewFolder[3] = "" ;  //contacts
        $$NewFolder[4] = "" ;  //select
        $$NewFolder[5] = "" ;
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drag.PortalID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$DraglistPortalID = "" ;
        $$Dontchange = ""
        ] ; $ac ) ]
End If
#create and add to folder
Perform Script [ “NewFolder [$$NewFolder or Menu] ”; Parameter: Let( [
    $$NewFolder[1] = "Folder" ;
    $$NewFolder[2] = window::xSys_ImportOptions[3] ;
    $$NewFolder[3] = $acimport ;
    $$NewFolder[4] = "" ;
    $$NewFolder[5] = ""
    ];"" ) ]
#create notes
Set Variable [ $end; Value:Let(   [
    $LL = $acimport  ;
    $count = 0  ;
    $lltemp = window::xPt_ItemIDForScript
    ];
    WordCount($LL)  ) ]
Loop
    Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
    Set Variable [ $link; Value: MiddleWords($LL; $count ; 1 ) ]
    Perform Script [ “LinkItems [$$drop.id[1], $$draglist[2] ] / NewLinks [ $$drop.id [1] ]”; Parameter: Let( [ $$drop.id = $link; $$draglist =
        "" ] ; "") ]
    Set Field [ window::xPt_ItemIDForScript; $link ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    If [ not IsEmpty( window::xSys_ImportOptions[2] ) ]
        Set Field [ window::xSys_NewItem; "New Note" ]
        Set Field [ window::xSys_NewItem[2]; $link   //Items to link to ]
        Set Field [ window::xSys_NewItem[3]; $link  //primaryID ]
        Set Field [ window::xSys_NewItem[4]; "external"   //portal name or "external" ]
        Set Field [ window::xSys_NewItem[5]; window::xSys_ImportOptions[2]   //note ]
        Perform Script [ “New [ [1] type, [2] link to, [3] primaryID, [4] Portal, [5] Note, [6] Ref ID]” ]
    End If
    If [ not IsEmpty(item.contact::Import Note 3) ]
        Set Field [ window::xSys_NewItem; "New Note" ]
        Set Field [ window::xSys_NewItem[2]; $link   //Items to link to ]
        Set Field [ window::xSys_NewItem[3]; $link  //primaryID ]
        Set Field [ window::xSys_NewItem[4]; "external"   //portal name or "external" ]
        Set Field [ window::xSys_NewItem[5]; item.contact::Import Note 3  //note ]
        Perform Script [ “New [ [1] type, [2] link to, [3] primaryID, [4] Portal, [5] Note, [6] Ref ID]” ]
        Set Field [ item.contact::Import Note 3; "" ]
    End If
    If [ not IsEmpty(item.contact::Import Note 2) ]
        Set Field [ window::xSys_NewItem; "New Note" ]
        Set Field [ window::xSys_NewItem[2]; $link   //Items to link to ]
        Set Field [ window::xSys_NewItem[3]; $link  //primaryID ]
        Set Field [ window::xSys_NewItem[4]; "external"   //portal name or "external" ]
        Set Field [ window::xSys_NewItem[5]; item.contact::Import Note 2  //note ]
        Perform Script [ “New [ [1] type, [2] link to, [3] primaryID, [4] Portal, [5] Note, [6] Ref ID]” ]
        Set Field [ item.contact::Import Note 2; "" ]
    End If
    If [ not IsEmpty(item.contact::Import Note 1) ]
        Set Field [ window::xSys_NewItem; "New Note" ]
        Set Field [ window::xSys_NewItem[2]; $link   //Items to link to ]
        Set Field [ window::xSys_NewItem[3]; $link  //primaryID ]
        Set Field [ window::xSys_NewItem[4]; "external"   //portal name or "external" ]
        Set Field [ window::xSys_NewItem[5]; item.contact::Import Note 1  //note ]
        Perform Script [ “New [ [1] type, [2] link to, [3] primaryID, [4] Portal, [5] Note, [6] Ref ID]” ]
        Set Field [ item.contact::Import Note 1; "" ]
    End If
    If [ Left( window::sys.GeneralDialogAnswers ; 1 )  = "y" ]
        Set Field [ window::xSys_NewItem; "New Note" ]
        Set Field [ window::xSys_NewItem[2]; $link   //Items to link to ]
        Set Field [ window::xSys_NewItem[3]; $link  //primaryID ]
        Set Field [ window::xSys_NewItem[4]; "external"   //portal name or "external" ]
        Set Field [ window::xSys_NewItem[5]; item.contact::Name & ", " & item.contact::Phone & ", "  & item.contact::Email ]
        Perform Script [ “New [ [1] type, [2] link to, [3] primaryID, [4] Portal, [5] Note, [6] Ref ID]” ]
        Set Field [ item.contact::Name; LeftWords( item.contact::Name ; 2 ) ]
        Set Field [ item.contact::Email; LeftWords( item.contact::Email ; 2 ) ]
        Set Field [ item.contact::Phone Area Code; Left( item.contact::Phone ; 3) ]
        Set Field [ item.contact::Phone; Substitute ( Replace(  LeftWords( item.contact::Phone ; 1 ) ; 1 ; 4 ; "") ; ["." ; "-" ] ) ]
    End If
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
End Loop
#
#import from notes database
Show Custom Dialog [ Title: "Import Note"; Message: "Are there notes to import?¶¶(the owner ID must be imported to item.note::
    xxIDL_LinkPrimaryItem)"; Buttons: “OK”, “Cancel”; Input #1: window::xSys_ImportOptions, "Import Note Field"; Input #2: window::
    xSys_ImportOptions[2], "Create note in notepad"; Input #3: window::xSys_ImportOptions[3], "Folder Name" ]
If [ Get ( LastMessageChoice ) = 1 ]
    Set Variable [ $acimport; Value:"" ]
    New Window [ Name: "~temp~"; Height: 80; Width: 80; Top: -400 ]
    Go to Layout [ “data.note” (item.note) ]
    Import Records [ Target: “item.note”; Method: Add; Character Set: “Mac Roman”; Field Mapping:
        Source field 2 import to item.note::xSys_Type
        Source field 3 import to item.note::xxCF_CreationTime
        Source field 7 import to item.note::xxIDL_Owner
        Source field 8 import to item.note::Note
        Source field 9 import to item.note::xxIDL_IDNumberInternal
        Source field 11 import to item.note::Date
        Source field 12 import to item.note::Email Subject
        Source field 13 import to item.note::Email Sender Name
        Source field 14 import to item.note::Email Sender Email
        Source field 15 import to item.note::Email cc
        Source field 16 import to item.note::xID_TemplatePaste
        Source field 17 import to item.note::xSys_EmailccChoice
        Source field 18 import to item.note::xID_Template
        Source field 19 import to item.note::xxIDL_LinkPrimaryItem
        Source field 20 import to item.note::Letter Name To Print
        Source field 21 import to item.note::Letter Signature
        Source field 22 import to item.note::Letter Reference
        Source field 25 import to item.note::File Container
        Source field 26 import to item.note::Email Recipient List
        Source field 27 import to item.note::Email cc List
        Source field 28 import to item.note::xxIDL_Trash
        Source field 29 import to item.note::xxIDL_IDNumberInternalSerialNumber
        Source field 30 import to item.note::xSys_EmailType
        Source field 31 import to item.note::xSys_TimeCreated
        Source field 32 import to item.note::Email Status
        Source field 34 import to item.note::File Keywords
        Source field 35 import to item.note::xxCF_Modified
        Source field 37 import to item.note::Email bcc
        Source field 39 import to item.note::xID_UserNoteID
        Source field 41 import to item.note::xID_TemplateName
        Source field 43 import to item.note::xID_UserNoteName
        Source field 45 import to item.note::product vendor
        Source field 46 import to item.note::product web
        Source field 47 import to item.note::product model
        Source field 48 import to item.note::product description
        Source field 49 import to item.note::product model web
        Source field 50 import to item.note::product category
        Source field 51 import to item.note::product model key
        Source field 52 import to item.note::product notes
        Source field 53 import to item.note::product key
        Source field 54 import to item.note::product web load
        Source field 59 import to item.note::xID_UserWebSiteID
        Source field 60 import to item.note::WebURL
        Source field 61 import to item.note::WebPassword
        Source field 62 import to item.note::WebImage
        Source field 63 import to item.note::WebAccount
        Source field 64 import to item.note::WebType
        Source field 65 import to item.note::WebName
        Source field 66 import to item.note::WebLastAccess
        Source field 69 import to item.note::xSys_AbreviateNote ]
    Replace Field Contents [ item.note::xxIDL_IDNumberInternal; Replace with calculation: item.note::xxIDL_IDNumberInternal & Let
```

```
            ($acimport = $acimport  & "¶"  & item.note::xxIDL_IDNumberInternal; "") ]
            [ No dialog ]
        Close Window [ Current Window ]
        Set Variable [ $acimport; Value: Substitute( Trim( Substitute( $acimport ; [" "; "@" ] ; ["¶" ; " " ] ) ) ; [" "; "¶" ] ; [ "@" ; " "] ; ["¶¶"; "¶" ]) ]
        If [ WordCount($acimport) > 0 ]
            Set Variable [ $end; Value:Let(   [
                $LL = $acimport  ;
                $count = 0  ;
                $lltemp = window::xPt_ItemIDForScript
                ];
                WordCount($LL)  ) ]
            Loop
                Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
                Set Variable [ $link; Value: MiddleWords($LL; $count ; 1 ) ]
                Perform Script [ “LinkItems [$$drop.id[1], $$draglist[2] ] / NewLinks [ $$drop.id [1] ]”; Parameter: Let( [ $$drop.id = $link; $
                    $draglist = "" ] ; "") ]
                Commit Records/Requests
                    [ Skip data entry validation; No dialog ]
                Perform Script [ “LinkItems [$$drop.id[1], $$draglist[2] ] / NewLinks [ $$drop.id [1] ]”; Parameter: Let( [ $$drop.id = $link; $
                    $draglist = item.note::xxIDL_LinkPrimaryItem ] ; "") ]
            End Loop
        End If
End If
#
Set Field [ window::xPt_ItemIDForScript; $lltemp ]
Exit Script [ Result: Let( [
        $$NewFolder[1] = "" ;  //type
        $$NewFolder[2] = "" ;  //name
        $$NewFolder[3] = "" ;  //contacts
        $$NewFolder[4] = "" ;  //select
        $$NewFolder[5] = "" ;
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drag.PortalID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$DraglistPortalID = "" ;
        $$Dontchange = ""
        ] ; $ac ) ]
```

Open and maintenance: temp import

```
Replace Field Contents [ item.note::xxIDL_IDNumberInternal; Replace with calculation: item.note::xxIDL_IDNumberInternal & Let($
    $acimport = $$acimport  & "¶"  & item.note::xxIDL_IDNumberInternal; "") ]
    [ No dialog ]
Go to Layout [ “Main Window” (window) ]
Show Custom Dialog [ Message: WordCount($$acimport); Buttons: “OK”, “Cancel” ]
Set Variable [ $$acimport; Value: Substitute( Trim( Substitute( $$acimport ; [" "; "@" ] ; ["¶" ; " " ] ) ) ; [" "; "¶" ] ; [ "@" ; " "] ; ["¶¶"; "¶" ]) ]
If [ WordCount($$acimport) > 0 ]
    Set Variable [ $end; Value:Let(   [
        $LL = $$acimport   ;
        $count = 0  ;
        $lltemp = window::xPt_ItemIDForScript
        ];
        WordCount($LL)  ) ]
    Loop
        Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
        Set Variable [ $link; Value: MiddleWords($LL; $count ; 1 ) ]
        Perform Script [ “LinkItems [$$drop.id[1], $$draglist[2] ] / NewLinks [ $$drop.id [1] ]”; Parameter: Let( [ $$drop.id = $link; $
            $draglist = "" ] ; "") ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
        Perform Script [ “LinkItems [$$drop.id[1], $$draglist[2] ] / NewLinks [ $$drop.id [1] ]”; Parameter: Let( [ $$drop.id = $link; $
            $draglist = item.note::xxIDL_LinkPrimaryItem ] ; "") ]
    End Loop
End If
```

Open and maintenance: Export Todo List

```
Freeze Window
Exit Script [  ]
If [ Get( AccountName ) = "JAS" ]
    Set Error Capture [ On ]
    Set Field [ window::xPt_ItemIDForScript; $$inboxID ]
    Set Field [ window::xPt_ItemIDForScript; item.folder::xxIDL_LinkListCalc ]
    Set Field [ window::xSys_TodoExportList; List ( portal.ToDoexport::link.todolist.export ) ]
    Set Field [ window::xPt_ItemIDForScript; "" ]
    If [ IsEmpty(window::xID_SpotlightEntry) ]
        Show All Records
        Omit Record
        Show Omitted Only
        Export Records [ File Name: "file:../../System Stuff/FMTodoList.txt"; Character Set: "Macintosh"; Field Order: window::
            xSys_TodoExportList ]
            [ No dialog ]
    End If
    #
    Exit Script [  ]
    Export Field Contents [ window::xSys_TodoExportList; "file:../../Documents/System Stuff/FMTodoList.txt" ]
    Show Custom Dialog [ Message: window::xPt_ItemIDForScript; Buttons: "OK", "Cancel" ]
End If
```

Open and maintenance: Old: Find (fix)

```
Set Error Capture [ On ]
Allow User Abort [ Off ]
Set Variable [ $$er; Value:"" ]
If [ PatternCount(Get ( LayoutName ) ; "Contact card") = 0 ]
    Set Variable [ $close; Value:"yes" ]
    New Window [ Top: Get(WindowTop) + 25; Left: Get(WindowLeft) + 45 ]
    Go to Layout [ “Main Window” (window) ]
End If
Show/Hide Status Area
    [ Hide ]
Enter Find Mode [  ]
Go to Object [ Object Name: "field.company" ]
Loop
    #enter find mode
    Enter Find Mode [  ]
        [ Pause ]
    Set Field [ window.contact::Status; "Active" ]
    Perform Find [  ]
    Exit Loop If [ Case(
        $$er = 1000;1;
         (Let($$er =Get(LastError) ; $$er = 0  or $$er = 400  ));1;
        ""
        ) ]
    #try switching name to company
    Modify Last Find
    Cut [ window.contact::Company Name ]
        [ Select ]
    Paste [ window.contact::Name ]
        [ Select ]
    Set Field [ window.contact::Status; "Active" ]
    Perform Find [  ]
    Exit Loop If [ (Let($$er =Get(LastError) ; $$er = 0  or $$er = 400  )) ]
    #try mixing contact and company
    Modify Last Find
    Set Field [ window.contact::Name; LeftWords(window.contact::Name;  1) ]
    Set Field [ window.contact::Company Name; RightWords(window.contact::Name; WordCount(window.contact::Name) - 1) ]
    Set Field [ window.contact::Status; "Active" ]
    Perform Find [  ]
    Perform Find/Replace [  ]
    Exit Loop If [ (Let($$er =Get(LastError) ; $$er = 0  or $$er = 400  )) ]
    #find failed, try again
    Beep
End Loop
If [ $$er =0  //Find worked ]
Else
    If [ $close = "yes" ]
        Close Window [ Current Window ]
    Else
        Enter Browse Mode
    End If
End If
#
Exit Script [  ]
#
#put before each find for initials filter, make new script if needed
Go to Record/Request/Page
    [ First ]
If [
            PatternCount(Get(CurrentPrivilegeSetName);"superuser")

             or
            window.contact::<Field Missing> =  "all" ]
Else
      New Record/Request
      Set Field [ window.contact::xxIDL_Owner; "<" & window.contact::<Field Missing> ]
      Omit Record
      New Record/Request
      Set Field [ window.contact::xxIDL_Owner; ">" & window.contact::<Field Missing> ]
      Omit Record
End If
```

Open and maintenance: Old: CompanyImage [copy image] FIX

```
Set Error Capture [ On ]
Freeze Window
Set Variable [ $a; Value:Let( [
          $ac = window.contact::xxIDL_IDNumberInternal;
          $sp = Get(ScriptParameter);
          $mk = Get ( ActiveModifierKeys )
          ];"")

          /*
          Shift = 1,
          Caps Lock = 2,
          Ctrl (Windows) and Control (Mac OS) = 4,
          Alt (Windows) and Option (Mac OS) = 8,
          ⌘ (Mac OS) = 16
          */ ]
If [ $sp = "copy image" ]
          Show Custom Dialog [ Title: "Copy Image"; Message: "Copy image from current contact to all contacts.¶Replace existing images?";
                    Buttons: “Copy”, “Copy/Replace”, “Cancel” ]
          If [ Get ( LastMessageChoice ) = 3 ]
                    Exit Script [  ]
          End If
          Set Field [ window::sys.temp image; window.contact::Image ]
          Go to Related Record [ From table: “portal.CompanyList”; Using layout: <Current Layout> ]
                    [ Show only related records; New window ]
          If [ Get(LastError)  =  0 and Get ( FoundCount ) < 150 ]
                    Replace Field Contents [ window.contact::Image; Replace with calculation: Case(
                              Get ( LastMessageChoice ) = 2; window::sys.temp image;
                              If(IsEmpty(window.contact::Image); window::sys.temp image ;window.contact::Image)
                              ) ]
                              [ No dialog ]
                    Close Window [ Current Window ]
          End If
End If
Commit Records/Requests
          [ Skip data entry validation; No dialog ]
```

Open and maintenance: Old: Clean up fields (fix)

```
Set Variable [ $n; Value:Get(RecordNumber) ]
Show Custom Dialog [ Title: "Message"; Message: "Clean up fields for"; Buttons: “This Record”, “Current Set”, “Cancel” ]
If [ Get(LastMessageChoice) = 1 ]
    Set Field [ window.contact::Phone Area Code; TextFormatRemove ( window.contact::Phone Area Code  ) ]
    Set Field [ window.contact::Phone2 Area Code; TextFormatRemove (  window.contact::Phone2 Area Code ) ]
    Set Field [ window.contact::<Field Missing>; TextFormatRemove (  window.contact::<Field Missing>
        ) ]
    Set Field [ window.contact::<Field Missing>; TextFormatRemove (  window.contact::<Field Missing>
        ) ]
    Set Field [ window.contact::<Field Missing>; TextFormatRemove (  window.contact::<Field Missing> ) ]
    Set Field [ window.contact::<Field Missing>; TextFormatRemove (  Upper(window.contact::<Field Missing> )) ]
    Set Field [ window.contact::<Field Missing>; TextFormatRemove (  window.contact::<Field Missing> ) ]
    Set Field [ window.contact::Company Name; TextFormatRemove (  window.contact::Company Name) ]
    Set Field [ window.contact::Department; TextFormatRemove ( window.contact::Department) ]
    Set Field [ window.contact::Email; TextFormatRemove ( window.contact::Email ) ]
    Set Field [ window.contact::Fax AreaCode; TextFormatRemove ( window.contact::Fax AreaCode ) ]
    Set Field [ window.contact::Fax Number; TextFormatRemove ( window.contact::Fax Number) ]
    Set Field [ window.contact::Name; TextFormatRemove ( window.contact::Name) ]
    Set Field [ window.contact::Phone; TextFormatRemove ( window.contact::Phone) ]
    Set Field [ window.contact::Phone2; TextFormatRemove (  window.contact::Phone2 ) ]
    Set Field [ window.contact::Ship Address; TextFormatRemove ( window.contact::Ship Address) ]
    Set Field [ window.contact::Ship Attention; TextFormatRemove (
        window.contact::Ship Attention) ]
    Set Field [ window.contact::Ship City; TextFormatRemove (
        window.contact::Ship City
        ) ]
    Set Field [ window.contact::Ship State; TextFormatRemove ( Upper(
        window.contact::Ship State)) ]
    Set Field [ window.contact::Ship Zip; TextFormatRemove (
        window.contact::Ship Zip) ]
    Set Field [ window.contact::Title; TextFormatRemove (
        window.contact::Title) ]
    Set Field [ window.contact::User Field; TextFormatRemove (
        window.contact::User Field) ]
    Set Field [ window.contact::Web Address; TextFormatRemove (
        window.contact::Web Address) ]
    Halt Script
End If
If [ Get(LastMessageChoice) = 2 ]
    Freeze Window
    Go to Record/Request/Page
        [ First ]
    Loop
        Set Field [ window.contact::Phone Area Code; TextFormatRemove ( window.contact::Phone Area Code  ) ]
        Set Field [ window.contact::Phone2 Area Code; TextFormatRemove (  window.contact::Phone2 Area Code ) ]
        Set Field [ window.contact::<Field Missing>; TextFormatRemove (  window.contact::<Field Missing>
            ) ]
        Set Field [ window.contact::<Field Missing>; TextFormatRemove (  window.contact::<Field Missing>
            ) ]
        Set Field [ window.contact::<Field Missing>; TextFormatRemove (  window.contact::<Field Missing> ) ]
        Set Field [ window.contact::<Field Missing>; TextFormatRemove (  Upper(window.contact::<Field Missing> )) ]
        Set Field [ window.contact::<Field Missing>; TextFormatRemove (  window.contact::<Field Missing> ) ]
        Set Field [ window.contact::Company Name; TextFormatRemove (  window.contact::Company Name) ]
        Set Field [ window.contact::Department; TextFormatRemove ( window.contact::Department) ]
        Set Field [ window.contact::Email; TextFormatRemove ( window.contact::Email ) ]
        Set Field [ window.contact::Fax AreaCode; TextFormatRemove ( window.contact::Fax AreaCode ) ]
        Set Field [ window.contact::Fax Number; TextFormatRemove ( window.contact::Fax Number) ]
        Set Field [ window.contact::Name; TextFormatRemove ( window.contact::Name) ]
        Set Field [ window.contact::Phone; TextFormatRemove ( window.contact::Phone) ]
        Set Field [ window.contact::Phone2; TextFormatRemove (  window.contact::Phone2 ) ]
        Set Field [ window.contact::Ship Address; TextFormatRemove ( window.contact::Ship Address) ]
        Set Field [ window.contact::Ship Attention; TextFormatRemove (
            window.contact::Ship Attention) ]
        Set Field [ window.contact::Ship City; TextFormatRemove (
            window.contact::Ship City
            ) ]
        Set Field [ window.contact::Ship State; TextFormatRemove ( Upper(
            window.contact::Ship State)) ]
        Set Field [ window.contact::Ship Zip; TextFormatRemove (
            window.contact::Ship Zip) ]
        Set Field [ window.contact::Title; TextFormatRemove (
            window.contact::Title) ]
        Set Field [ window.contact::User Field; TextFormatRemove (
            window.contact::User Field) ]
        Set Field [ window.contact::Web Address; TextFormatRemove (
            window.contact::Web Address) ]
        Go to Record/Request/Page
            [ Next; Exit after last ]
    End Loop
    Go to Record/Request/Page [ $n ]
End If
```

Open and maintenance: Old: only one company name

```
Go to Record/Request/Page
    [ First ]
Set Variable [ $c; Value:window.contact::Company Name ]
Go to Record/Request/Page
    [ Next; Exit after last ]
Loop
    If [ window.contact::Company Name = $c ]
        Omit Record
    Else
        Go to Record/Request/Page
            [ Next; Exit after last ]
        Set Variable [ $c; Value:window.contact::Company Name ]
    End If
End Loop
```

Print FIX: To Print Menu

```
Enter Browse Mode
Set Field [ window.contact::<Field Missing>; Upper(window.contact::<Field Missing>) ]
Set Field [ window.contact::Ship State; Upper( window.contact::Ship State) ]
Set Field [ window.contact::xPrint_LabelStartNumber; "" ]
New Window [ Name: "Print"; Height: 668; Width: 806; Top: 180; Left: 285 ]
Go to Related Record [ From table: “window.contact”; Using layout: “Print” (window.contact) ]
    [ Show only related records ]
Go to Layout [ “Print” (window.contact) ]
Perform Script [ “WindowResize [centered, full, ]”; Parameter: "centered" ]
Set Field [ window.contact.print::sys.selected; If(IsEmpty(window.contact.print::sys.selected); window.contact.print::print.layout exact
    name;window.contact.print::sys.selected) ]
Perform Script [ “Select print layout/refresh picture”; Parameter: "refresh picture" ]
```

Print FIX: Print

```
Set Error Capture [ On ]
Set Variable [ $count; Value:Let($layout = window.contact.print::sys.selected;1) ]
New Window [ Height: 1; Width: 1; Top: -80 ]
Loop
    Go to Layout [ $count ]
    Exit Loop If [ Get(LayoutName) = $layout or $count > 1000 ]
    Set Variable [ $count; Value:$count + 1 ]
End Loop
If [ window.contact.print::sys.setting.all one  ≠  "Entire Set" ]
    Show All Records
    Omit Record
    Show Omitted Only
    Print [ Current record; All Pages; Orientation: Portrait; Paper size: 8.5" x 11" ]
        [ Restore ]
    Set Variable [ $error; Value:Get ( LastError ) ]
Else
    Print [ Records being browsed; All Pages; Orientation: Portrait; Paper size: 8.5" x 11" ]
        [ Restore ]
    Set Variable [ $error; Value:Get ( LastError ) ]
End If
If [ $error = 0 ]
    #make note here
End If
Close Window [ Current Window ]
#
```

Print FIX: Rebuild list

```
Freeze Window
Set Variable [ $count; Value:Let([$max = Get ( LayoutCount ); $temp = window.contact::xSys_PrintKeyChoice]; 1) ]
Loop
    Go to Layout [ $count ]
    Set Variable [ $layout; Value: Get(LayoutName) ]
    If [ Left($layout;3) = "CFM" or
        Left($layout;3) = "CLB" or
        Left($layout;3) = "CLS" ]
        Set Field [ window.contact::xSys_PrintKeyChoice; $layout ]
        Go to Layout [ <unknown> ]
        If [ $layout  ≠   window.contact.print::print.layout exact name ]
            Go to Layout [ “data.print” (window.contact.print) ]
            New Record/Request
            Set Field [ window.contact.print::print.layout exact name; $layout ]
        End If
        Go to Layout [ “data.print” (window.contact.print) ]
        Set Field [ window.contact.print::print.mark for delete; "" ]
    End If
    Exit Loop If [ Let( $count = $count + 1; $count) > $max

         or $count > 1000 ]
End Loop
Set Error Capture [ On ]
Go to Layout [ “data.print” (window.contact.print) ]
Enter Find Mode [  ]
Set Field [ window.contact.print::print.mark for delete; 1 ]
Perform Find [  ]
If [ Get(FoundCount) > 0 ]
    Delete All Records
        [ No dialog ]
End If
Show All Records
Replace Field Contents [ window.contact.print::print.mark for delete; Replace with calculation: 1 ]
    [ No dialog ]
Go to Layout [ original layout ]
Set Field [ window.contact::xSys_PrintKeyChoice; $temp ]
```

Print FIX: Select print layout/refresh picture

```
If [ Get ( ScriptParameter )  ≠  "refresh picture" ]
    Set Field [ window.contact.print::sys.setting.all one; Case(
        Left(window.contact.print::print.layout exact name;3) = "CFM"; "Current Record";
        "Entire Set"
            ) ]
    Set Field [ window.contact.print::sys.selected; window.contact.print::print.layout exact name ]
End If
Set Variable [ $count; Value:Let($layout = window.contact.print::sys.selected;1) ]
New Window [ Height: 1; Width: 1; Top: -80 ]
Loop
    Go to Layout [ $count ]
    Exit Loop If [ Get(LayoutName) = $layout or $count > 1000 ]
    Set Variable [ $count; Value:$count + 1 ]
End Loop
If [ window.contact.print::sys.setting.all one  ≠  "Entire Set" ]
    Show All Records
    Omit Record
    Show Omitted Only
End If
Enter Preview Mode
Copy [  ]
    [ Select ]
Close Window [ Current Window ]
Freeze Window
Paste [ window.contact.print::print.preview image ]
    [ Select ]
Go to Object [ Object Name: If(window.contact.print::sys.setting.portrait landscapre = "Landscape"; "tab.landscape";"tab.portrait") ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
```

Print FIX: toggle current record entire set

```
Set Field [ window.contact.print::sys.setting.all one; Case(
        window.contact.print::sys.setting.all one = "Current Record";"Entire Set";
        "Current Record"
              ) ]
Perform Script [ “Select print layout/refresh picture”; Parameter: "refresh picture" ]
```

Print FIX: toggle portrait landscape

```
Set Field [ window.contact.print::sys.setting.portrait landscapre; Case(
    window.contact.print::sys.setting.portrait landscapre = "Landscape";"Portrait";
    "Landscape"
        ) ]
If [ window.contact.print::sys.setting.portrait landscapre = "Landscape" ]
    Print Setup [ Orientation: Landscape; Paper size: 8.5" x 11" ]
        [ Restore; No dialog ]
Else
    Print Setup [ Orientation: Portrait; Paper size: 8.5" x 11" ]
        [ Restore; No dialog ]
End If
Perform Script [ “Select print layout/refresh picture”; Parameter: "refresh picture" ]
```

Print FIX: Print Label set order

```
Set Error Capture [ On ]
Go to Layout [ <unknown> ]
If [ window.contact::<Field Missing> = "Current Customer" ]
    Set Field [ window.contact::xPrint_LabelSortNumber; 1 ]
Else
    Go to Record/Request/Page
        [ First ]
    Go to Field [ window.contact::xPrint_LabelSortNumber ]
    Replace Field Contents [ window.contact::xPrint_LabelSortNumber; Replace with serial numbers: Custom values; Initial value: 1;
        Increment value: 1 ]
        [ No dialog ]
End If
Show Omitted Only
If [ GetAsNumber(window.contact::xPrint_LabelStartNumber) = 1 or GetAsNumber(window.contact::xPrint_LabelStartNumber) = 0 or
    IsEmpty(window.contact::xPrint_LabelStartNumber) ]
Else
    If [ Get(FoundCount) < (window.contact::xPrint_LabelStartNumber -1 ) ]
        Show Custom Dialog [ Title: "Message"; Message: "Can not process this print job.  Try using a new sheet with a start value of
            1"; Buttons: “Cancel” ]
        Go to Layout [ <unknown> ]
        Perform Script [ “<unknown>” ]
        Go to Field [ window.contact::xPrint_LabelSortNumber ]
        Replace Field Contents [ window.contact::xPrint_LabelSortNumber; Replace with calculation: "" ]
            [ No dialog ]
        Go to Layout [ original layout ]
        Halt Script
    End If
    Go to Record/Request/Page
        [ First ]
    Loop
        Set Field [ window.contact::xPrint_LabelSortNumber; 0 ]
        Go to Record/Request/Page
            [ Next; Exit after last ]
        Exit Loop If [ Get(RecordNumber) = window.contact::xPrint_LabelStartNumber ]
    End Loop
End If
Enter Find Mode [  ]
Insert Text [ window.contact::xPrint_LabelSortNumber; “=” ]
    [ Select ]
Omit Record
Perform Find [  ]
If [ Get(FoundCount) = 0 ]
    Show Custom Dialog [ Title: "Message"; Message: "Can not process this print job.  Try using a new sheet with a start value of 1";
        Buttons: “Cancel” ]
    Go to Layout [ <unknown> ]
    Perform Script [ “<unknown>” ]
    Go to Field [ window.contact::xPrint_LabelSortNumber ]
    Replace Field Contents [ window.contact::xPrint_LabelSortNumber; Replace with calculation: "" ]
        [ No dialog ]
    Go to Layout [ original layout ]
    Halt Script
End If
Sort Records [ Specified Sort Order: window.contact::xPrint_LabelSortNumber; ascending ]
    [ Restore; No dialog ]
Go to Record/Request/Page
    [ First ]
Go to Field [ window.contact::xPrint_LabelOn ]
Replace Field Contents [ window.contact::xPrint_LabelOn; Replace with calculation: window.contact::xPrint_LabelSortNumber ]
    [ No dialog ]
```

Go to Field [window.contact::xPrint_LabelSortNumber]
Replace Field Contents [window.contact::xPrint_LabelSortNumber; Replace with calculation: ""]
    [No dialog]
Go to Layout [original layout]

Preferences: relookup region

```
Set Error Capture [ On ]
Allow User Abort [ Off ]
Show Custom Dialog [ Title: "Message"; Message: "This will relculate the regions for all customers (except manually entered regions).
    This operation may take a while.  Continue?"; Buttons: “All Records”, “Found Set”, “Cancel” ]
If [ Get(LastMessageChoice) = 3 ]
    Exit Script [  ]
Else If [ Get(LastMessageChoice) = 1 ]
    Show All Records
End If
Go to Layout [ <unknown> ]
Go to Field [ window.contact::xID_Region ]
Relookup Field Contents [ window.contact::xID_Region ]
    [ No dialog ]
Go to Layout [ original layout ]
```

Preferences: Copy

```
Duplicate Record/Request
Set Field [ window.contact.user::user.first name; "" ]
Set Field [ window.contact.user::user.name; "" ]
Set Field [ window.contact.user::sys.initials.my initials; "" ]
Set Field [ window.contact.user::ID default; "" ]
Set Field [ window.contact.user::user.sys.commision; "" ]
Set Field [ window.contact.user::user.email; If(IsEmpty(window.contact.user::user.email);"";   Middle(window.contact.user::user.email;
        Position(window.contact.user::user.email; "@"; 1; 1);30)) ]
```

Preferences: delete

```
If [ window.contact.user::ID default = "default" ]
    Show Custom Dialog [ Title: "Message"; Message: "This record contains system data and can not be deleted"; Buttons: “Cancel” ]
    Exit Script [  ]
End If
Delete Record/Request
```

Email FIX: New email [cc, bcc]

```
If [ Get(ScriptParameter) = "cc" ]
        Set Field [ item.note::Email cc; Substitute (
                Case(
                 IsEmpty(item.note::xSys_EmailccChoice); item.note::Email cc ;
                 If ( IsEmpty ( item.note::Email cc ); item.note::xSys_EmailccChoice; item.note::Email cc & "¶" & item.note::
                xSys_EmailccChoice )
                )
                ; "¶"; ", ") ]
        Set Field [ item.note::xSys_EmailccChoice; "" ]
Else If [ Get(ScriptParameter) = "bcc" ]
        Set Field [ item.note::Email bcc; Substitute (
                Case(
                 IsEmpty(item.note::xSys_EmailccChoice[2]); item.note::Email bcc ;
                 If ( IsEmpty ( item.note::Email bcc ); item.note::xSys_EmailccChoice[2]; item.note::Email bcc & "¶" & item.note::
                xSys_EmailccChoice[2] )
                )
                ; "¶"; ", ") ]
        Set Field [ item.note::xSys_EmailccChoice[2]; "" ]
Else
        Set Field [ window::xSys_NewItem; "New Email" ]
        Set Field [ window::xSys_NewItem[2]; window::xPt_ContactID ]
        Set Field [ window::xSys_NewItem[3]; window::xPt_ContactID ]
        Set Field [ window::xSys_NewItem[4]; "external" ]
        Perform Script [ “New [ [1] type, [2] link to, [3] primaryID, [4] Portal, [5] Note, [6] Ref ID]” ]
        Commit Records/Requests
                [ Skip data entry validation; No dialog ]
End If
```

Email FIX: Send email

```
Allow User Abort [ Off ]
If [ IsEmpty( item.note::Email Recipient List ) ]
    Show Custom Dialog [ Title: "Message"; Message: "An email address must be present."; Buttons: “Cancel” ]
    Exit Script [  ]
End If
Set Field [ item.note::Email Status; "sent" ]
Set Field [ window.contact::Call Last Attempt; Get(CurrentDate) ]
Send Mail [ Send via E-mail Client; To: item.note::Email Recipient List; CC: item.note::Email cc; Subject: item.note::Email Subject;
    Message: item.note::Note & If(window.contact.user::user.sys.attach signature = "1"; window.contact.user::user.signature;"") ]
Perform AppleScript [ Calculated AppleScript: "set theSignature to \""   & window.user::user.email.signame & "\""   & "¶" &
    "tell application \"Mail\"" & "¶" &
        "activate" & "¶" &
        "tell application \"System Events\" to tell process \"Mail\"" & "¶" &
                "click pop up button 1 of window 1" & "¶" &
                "click menu item theSignature of menu 1 of pop up button 1 of window 1" & "¶" &
        "end tell" & "¶" &
    "end tell" ]
Close Window [ Current Window ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
```

Email FIX: email template [paste, select, new, delete, merge]

```
Freeze Window
Set Variable [ $sp; Value:Get(ScriptParameter) ]
Set Variable [ $pos; Value:Let( [
        $min = Min ( window.template.email.all::xID_Template );
        $max = Max ( window.template.email.all::xID_Template ) ;
        $item = window::xID_TemplateEmail ;
        $list = GetAsText( List( window.template.email.all::xID_Template  )  ) ;
        $PR = Position (  Filter( Substitute(  Substitute( "TTT" &  $list  & "TTT"; "¶" ; "TTT¶TTT" ) ; "TTT"  & $item  & "TTT" ; "#") ; "#¶"  ) ;
        "#" ;1; 1) ;
        $current = MiddleWords( $list; $PR; 1)

        ];
        $PR
        ) ]
If [ $sp = "new" ]
        Set Field [ window::xID_TemplateEmail; $max + 1 ]
        Commit Records/Requests
                [ Skip data entry validation; No dialog ]
        Set Field [ window.template.email::xID_TemplateName; "New Template" ]
        Commit Records/Requests
                [ Skip data entry validation; No dialog ]
        Go to Object [ Object Name: "field.template.name" ]
        Select All
Else If [ $sp = "delete" ]
        If [ Case(
                IsEmpty( window::xID_TemplateEmail ) ; "" ;
                window::xID_TemplateEmail  ≠  window.template.email::xID_Template ; "";
                1) ]
                Set Field [ window.template.email::xxIDL_IDNumberInternal; "delete" ]
                Set Field [ window.template.email::xID_Template; "delete" ]
                Commit Records/Requests
                        [ Skip data entry validation; No dialog ]
                Set Field [ window::xID_TemplateEmail; Max ( window.template.email.all::xID_Template ) ]
        End If
Else If [ $sp = "paste" ]
        Set Field [ window::xID_TemplateEmail; window.template.email.all::xID_Template ]
        Commit Records/Requests
                [ Skip data entry validation; No dialog ]
        Set Field [ item.note::Note; Let(  nn = Substitute(window.template.email::Note;
                    ["<<user first name>>";window.user::user.first name];
                    ["<<user name>>";window.user::user.name];
                    ["<<first name>>";  window.contact::First Name];
                    ["<<last name>>";  window.contact::Last Name];
                    ["<<name>>";  window.contact::Name];
                    ["<<company name>>";  window.contact::Company Name] );

                If(IsEmpty(item.note::Note); nn ;   item.note::Note &"¶¶"  & nn )
                ) ]
        Set Field [ item.note::Email Subject; Let(  nn = Substitute(window.template.email::Email Subject;
                    ["<<user first name>>";window.user::user.first name];
                    ["<<user name>>";window.user::user.name];
                    ["<<first name>>";  window.contact::First Name];
                    ["<<last name>>";  window.contact::Last Name];
                    ["<<name>>";  window.contact::Name];
                    ["<<company name>>";  window.contact::Company Name] );

                If(IsEmpty(item.note::Email Subject); nn ;   item.note::Email Subject )
                ) ]
        Go to Object [ Object Name: "tab.template.blank" ]
        Commit Records/Requests
                [ Skip data entry validation; No dialog ]
Else If [ $sp = "select" ]
        Set Field [ window::xID_TemplateEmail; window.template.email.all::xID_Template ]
Else If [ LeftWords( $sp ; 1)= "merge" ]
        If [ Get ( ActiveFieldName  ) = "Note" and Get ( ActiveFieldTableName ) = "window.template.email" ]
                Set Selection [ window.template.email::Note; Start Position: Get ( ActiveSelectionStart ) + Get ( ActiveSelectionSize ); End
                        Position: "" ]
        Set Selection [ window.template.email::Note; Start Position: Get ( ActiveSelectionStart ) + Get ( ActiveSelectionSize ); End
            Position: "" ]
        Insert Calculated Result [ " <<"  & MiddleWords($sp; 2; 4)   & ">> " ]
    End If
End If
```

Email FIX: paste email to reg [paste, clear]

```
Freeze Window
Set Error Capture [ On ]
Set Variable [ $a; Value:Let( [
      $sp = Get(ScriptParameter) ;
      $mk = Get ( ActiveModifierKeys )
      ];"") ]
If [ $sp = "paste" ]
      Paste [ item.registration::sys.lookup paste text ]
            [ Select; No style ]
End If
If [ $sp = "clear" ]
      Set Field [ item.registration::sys.lookup paste text; "" ]
End If
Commit Records/Requests
      [ Skip data entry validation; No dialog ]
```

Email FIX: goto mail find email address

```
If [ not IsEmpty( window.contact::xxIDL_IDNumberInternal ) and not IsEmpty( window.contact::Email ) ]
    Copy [ window.contact::Email ]
        [ Select ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Perform AppleScript [ Native AppleScript:
        tell application "Mail"
                activate
        tell application "System Events" to keystroke "f" using {command down, option down}
        tell application "System Events" to keystroke "f" using {command down, option down}
        tell application "System Events" to keystroke "v" using {command down}

        end tell
         ]
End If
```

New: New [[1] type, [2] link to, [3] primaryID, [4] Portal, [5] Note, [6] Ref ID]

```
Freeze Window
Set Variable [ $lltemp; Value:window::xPt_ItemIDForScript ]
Set Variable [ $$NewItem; Value:Case(
          IsEmpty( Trim ( window::xSys_NewMenuChoice )  ) ; $$NewItem[1] ;
          Trim ( window::xSys_NewMenuChoice )
          ) ]
#[1] TYPE, [2] LINK TO [3] PRIMARY ID [4] PORTAL NAME [5] NOTE [6] Reference ID [7] Reminder Date [8] Reminder Note, [9] Folder
          ID
#To use this script, simply changethe field window::xSys_NewItem[1-12], use the script "NewNote [type]" for notes
Set Field [ window::xSys_NewMenuChoice; " " ]
Set Variable [ $t; Value:$$NewItem[1] ]
Set Field [ window::xSys_NewItem; "" ]
Set Variable [ $$NewItem; Value:$t ]
Commit Records/Requests
          [ Skip data entry validation; No dialog ]
#
Loop
          Exit Loop If [ IsEmpty ( Trim( $$NewItem ) ) ]
          #Set Primary Contact
          Set Field [ window::xPt_ItemIDForScript; $$NewItem[3] ]
          Set Variable [ $PC; Value:Evaluate( join::xxIDL_FNLinkPrimaryItem ) ]
          #
          Set Field [ window::xPt_ItemIDForScript; Let( [
                    $link =
                    Case(
                    $$NewItem[1] = "New Email Received" ; "M";
                    $$NewItem[1] = "New Email" ; "M";
                    $$NewItem[1] = "New Communication" ; "Y";
                    $$NewItem[1] = "New Note" ; "N";
                    $$NewItem[1] = "New Missed" ; "X";
                    $$NewItem[1] = "New VM" ; "V";
                    $$NewItem[1] = "New Reg. History" ; "V";
                    $$NewItem[1] = "New Print" ; "Z" ;

                    $$NewItem[1] = "New Opportunity" ; "AO";
                    $$NewItem[1] = "New Prospect" ; "AP";
                    $$NewItem[1] = "New To Do" ; "AT";
                    $$NewItem[1] = "New Call" ; "AC";
                    $$NewItem[1] = "New Appointment" ; "AS";
                    $$NewItem[1] = "New Scheduled" ; "AS";

                    $$NewItem[1] = "New Registration" ; "R";

                    $$NewItem[1] = "New File" ; "D";

                    $$NewItem[1] = "New Quote" ; "Q";
                    $$NewItem[1] = "New Order" ; "O";
                    $$NewItem[1] = "New PO" ; "P";

                    $$NewItem[1] = "New Contact" ; "C";
                    $$NewItem[1] = "New Contact Same Company" ; "C";
                    $$NewItem[1] = "Same Contact New Company" ; "C";
                    $$NewItem[1] = "New Contact Same Company As Selected" ; "C";


                    "N"
                    )   &
                    Right("00" & Day(Get(CurrentDate));2)  &
                    Right("00" & Month(Get(CurrentDate));2) &
                    Right(Year(Get(CurrentDate));2)  &
                    Right("00000" & GetAsText(Get(RecordID));5) &
                    Right("0000" & GetAsText(Int(9999*Random));4)
                    ]; $link ) ]
          Set Field [ join::xxIDL_IDNumberPortal; $link ]
          Commit Records/Requests
                    [ Skip data entry validation; No dialog ]
          Go to Related Record [ From table: “join”; Using layout: “data.join” (join) ]
                    [ Show only related records; New window ]
          Set Field By Name [ join::xxIDL_FNIDNumberInternal; $link ]
          Set Field By Name [ If (
                    Left( $link; 1)  ≠ "C"; join::xxIDL_FNLinkPrimaryItem ;
                    ""); $PC ]
          Commit Records/Requests
                    [ Skip data entry validation; No dialog ]
          If [ $$NewItem[1] = "New File" ]
                    Insert File [ item.note::File Container ]
                        [ Reference ]
                    If [ Get ( LastError )  ≠ 0 ]
                        Delete Record/Request
                            [ No dialog ]
                        Close Window [ Current Window ]
                        Exit Loop If [ 1 ]
                    End If
          End If
          Close Window [ Current Window ]
          Set Field [ window::xID_ReminderID; $link ]
          Set Field [ window.reminder::Date; GetAsDate(  $$NewItem[7] ) ]
          Set Field [ window.reminder::Note; $$NewItem[8] ]
          #
          #link
          Perform Script [ “LinkItems [$$drop.id[1], $$draglist[2] ] / NewLinks [ $$drop.id [1] ]”; Parameter: Let( [$$drop.id = $link; $$draglist  =
                    $$NewItem[2]] ; "" ) ]
          Set Field [ window::xPt_ItemIDForScript; $link ]
          #move to inbox ($$draglist becomes $link or "")
          Perform Script [ “Move/Copy/ClearToAnyFolder [ $$drop.ID [1], $$DragList [2] cleans, g(sp) = from folders ] ”; Parameter: Let( [

                    $$drop.ID =
                    Case(
                    not IsEmpty( $$NewItem[9]) ; $$NewItem[9] ;
                    PatternCount( window.user::xPrefUser_CopyToInboxWhenCreated; join::link.type.detailed ) > 0 ; $$inboxID ;
                    "" ) ;

                    $$DragList =
                    If( PatternCount( window.user::xPrefUser_CopyToInboxWhenCreated; join::link.type.detailed ) > 0 or not IsEmpty( $$NewItem
                    [9] );$link ;
                    "" )

                     ] ; "") ]
          Set Field [ window::xPt_ItemIDForScript; $link ]
          #item specific
          Exit Loop If [ $NewItemType = "New File" ]
          If [ $$NewItem[1] = "New Quote" ]
                    Set Field [ item.quote.revision::System rev Quote date; Get(CurrentDate) ]
          Else If [ $$NewItem[1] = "New Contact" or
                    $$NewItem[1] = "Same Contact New Company" or
                    $$NewItem[1] = "New Contact Same Company" ]
                    If [ $$NewItem[1] = "Same Contact New Company" ]
                        Perform Script [ “SameContactNewCompany [$ac (old), link (new)]”; Parameter:  $$NewItem[6]  & " "  & $link ]
                    Else If [ $$NewItem[1] = "New Contact Same Company" ]
                        Perform Script [ “NewContactSameCompany [$ac (old), link (new)]”; Parameter: Case(
                            $$NewItem[1] = "New Contact Same Company" ; $$NewItem[6];
                            $$NewItem[1] = "New Contact Same Company As Selected"  and Left( window::
                            xPt_SelectedItems__CurrentClickedLink ; 1 ) = "C" ; window::xPt_SelectedItems__CurrentClickedLink;
                            "")
                             & " "  & $link ]
                    Else If [ $$NewItem[1] = "New Contact" ]
                    End If
                    If [ window::xPt_MainListLayout  =  "importemail" or
                        window::xPt_MainListLayout  = "ResearchCard" ]
                        Set Field [ window::xPt_ContactID; $link ]
                    Else
                        Perform Script [ “DoubleClicked [ link (optional), $$EnterNote , $$drag.portal]”; Parameter: $link ]
          End If
                    Exit Loop If [ 1 ]
          Else If [ $$NewItem[1] = "New Email" ]
                    Perform Script [ “WindowNew [ main, folder, folder options, pref, note, newemail, importemail, phonerecord, websites]”;
                        Parameter: "newemail" ]
                    Set Field [ window::xPt_ContactID; $PC ]
                    Set Field [ window::xPt_ItemID; $link ]
                    Set Field [ item.note::Email Recipient List; window.contact::Email ]
                    Set Field [ item.note::Email Sender Email; window.user::user.email ]
                    Set Field [ item.note::Email Sender Name; window.user::user.name ]
                    Set Field [ item.note::Email Status; "Unsent" ]
                    Go to Object [ Object Name: "field.emailsubject" ]
                    Exit Loop If [ 1 ]
          Else If [ $$NewItem[1] = "New Email Received" or
                    $$NewItem[1] = "New Email"  or
                    $$NewItem[1] = "New Communication"  or
                    $$NewItem[1] = "New Note"  or
                    $$NewItem[1] = "New Missed"  or
                    $$NewItem[1] = "New VM"  or
                    $$NewItem[1] = "New Reg. History"  or
                    $$NewItem[1] = "New Print" ]
                    Set Field [ item.note::Note; $$NewItem[5] ]
                    Exit Loop If [ 1 ]
          Else If [ Left($link ;1 )  = "A" ]
                    Set Field [ item.action::Note; Case(
                        Trim ( $$NewItem[5] )  ≠ Trim( $$NewItem[1] ) ; $$NewItem[5] ;
                        ""
                        ) ]
                    Perform Script [ “DoubleClicked [ link (optional), $$EnterNote , $$drag.portal]”; Parameter: $link ]
          End If
          If [ Get(LayoutName)  =  "Email Import" ]
                    Set Field [ window::xID_EmailimportRegistration; Case(
                        Left( $link ;1 ) = "R" ; $Link ; window::xID_EditLinkRegistration ) ]
                    Set Field By Name [ Case(
                        Left( $link ;1 ) = "R" ; "window.registration::sys.lookup paste text" ; "" ); item.note::Note ]
                    Go to Object [ Object Name: Case(
                        GetLayoutObjectAttribute("tab.mainlist 3";"isFrontTabPanel")  =  1 ; "field.contact.name 3" ;
                        GetLayoutObjectAttribute("tab.mainlist 2";"isFrontTabPanel")  =  1 ; "field.contact.name 2" ;
                        "field.contact.name"
                        ) ]
                    Exit Loop If [ 1 ]
          End If
          Commit Records/Requests
                    [ Skip data entry validation; No dialog ]
          Exit Loop If [ $$NewItem[4] = "external" ]
          #
          #open
          Perform Script [ “DoubleClicked [ link (optional), $$EnterNote , $$drag.portal]”; Parameter: $link ]
          If [ 1 = 2 ]
                    Perform Script [ “ScrollToItemAndSelect [ link, $$drag.portal ]”; Parameter: $link ]
                    Set Variable [ $$EnterNote; Value:Case(
                        Get(ScriptResult ) = "Failed" ; "key" ;
                        Filter (Left( $link ; 1)  ; "AXVYNZ"  ) > 0; "Enter";
                        "key"
                        ) ]
          End If
          Exit Loop If [ 1 ]
      End Loop
      #
      Loop
          Exit Loop If [ Let( $count = $count + 1 ; $count)   >  12 ]
          Set Field [ window::xSys_NewItem[$count]; "" ]
```

```
End Loop
Set Field [ window::xSys_NewItem[12]; $link ]
Set Field [ window::xPt_ItemIDForScript; $lItemp ]
Exit Script [ Result: Let( [
        $$EnterNote = "" ;
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.portalid = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$Dontchange = ""
        ] ; "" ) ]
Show Custom Dialog [ Message: "1 "  & window::xSys_NewItem[ 1 ]  & "¶" &
        "2 "  & window::xSys_NewItem[ 2 ]  & "¶" &
        "3 "  & window::xSys_NewItem[ 3 ]  & "¶" &
        "4 "  & window::xSys_NewItem[ 4 ]  & "¶" &
        "5 "  & window::xSys_NewItem[ 5 ]  & "¶" &
        "6 "  & window::xSys_NewItem[ 6 ]  & "¶¶" &
        "1 $$"  & $$NewItem[1]  & "¶" &
        "2 $$"  & $$NewItem[2] & "¶" &
        "3 $$"  & $$NewItem[3]  & "¶" &
        "4 $$"  & $$NewItem[4]  & "¶" &
        "5 $$"  & $$NewItem[5]  & "¶"&
        "6 $$"  & $$NewItem[6]  & "¶"  &
        "Menu "  & window::xSys_NewMenuChoice  & "¶"; Buttons: “OK”, “Cancel” ]
```

New: NewNote [type]

```
Freeze Window
#To use this script, simply changethe field window::xSys_NewItem[1-12]
#[1] TYPE, [2] ITEMS TO LINK TO $$DragList, [3] PRIMARY ID $$drag.portalID, [4] PORTAL NAME (OR EXTERNAL) $$drag.portal, [5]
        NOTE or $$drag.note, [6] Reference ID
Commit Records/Requests
        [ Skip data entry validation; No dialog ]
If [ Get(ScriptParameter ) = "Left VM" or
        Get(ScriptParameter ) = "Not In" or
        Get(ScriptParameter ) = "Different VM" or
        Get(ScriptParameter ) = "Left VM" or
        Get(ScriptParameter ) = "Wrong Number" or
        Get(ScriptParameter ) = "No Answer"  or
        Get(ScriptParameter ) = "Non Working number" ]
        Perform Script [ “Dial [ dtmf, phonefield, hangup, switch, dbn (first, last, selected) ]”; Parameter: "hangup" ]
End If
Set Field [ window::xSys_NewNoteType; Case(
        not IsEmpty( Get(ScriptParameter) ) ; Get(ScriptParameter) ;
        IsEmpty ( window::xSys_NewNoteType[ 1 ]  ) ; "Communication" ;
        window::xSys_NewNoteType ) ]
#
Set Variable [ $pos; Value:Let( [
        $item = Trim(window::xSys_NewNoteType[1]) ;
        $list =  List(window.user::xNote_Note) ;
        $PR = Position (  Filter( Substitute(  Substitute( "TTT" &  $list  & "TTT"; "¶" ; "TTT¶TTT" ) ; "TTT"  & $item  & "TTT" ; "#$#") ; "#
        $#¶"  ) ; "#$#" ;1; 1)
        ];
        $PR
        ) ]
If [ $pos > 0 ]
        Set Field [ window.contact::Status; Case(
                IsEmpty( window.user::xNote_NoteChangeStatus[ $pos ] ) ; window.contact::Status ;
                window.user::xNote_NoteChangeStatus[ $pos ]
                ) ]
        Set Field [ window.contact::Status Reason; Case(
                IsEmpty( window.user::xNote_NoteChangeStatusReason[ $pos ] ) ; window.contact::Status Reason ;
                window.user::xNote_NoteChangeStatusReason[ $pos ]
                ) ]
        Set Field [ window.contact::Call Category; Case(
                IsEmpty( window.user::xNote_NoteChangeCallCategory[ $pos ] ) ; window.contact::Call Category ;
                window.user::xNote_NoteChangeCallCategory[ $pos ]
                ) ]
        Set Field [ window.contact::Role; Case(
                IsEmpty( window.user::xNote_NoteChangeRole[ $pos ] ) ; window.contact::Role ;
                window.user::xNote_NoteChangeRole[ $pos ]
                ) ]
        Set Field [ window.contact::Call Last Attempt; Case(
                IsEmpty( window.user::xNote_NoteCallAttempt[ $pos ] ) ; window.contact::Call Last Attempt ;
                Get(CurrentDate)
                ) ]
        Set Field [ window.contact::Call Last Connect; Case(
                IsEmpty( window.user::xNote_NoteCallConnect[ $pos ] ) ; window.contact::Call Last Connect ;
                Get(CurrentDate)
                ) ]
        Set Field [ window.contact::Call Next Connect Manual; Case(
                IsEmpty( window.contact::Call Next Connect Manual ) ; "" ;
                IsEmpty( window.user::xNote_NoteCallConnect[ $pos ] ) ; window.contact::Call Next Connect Manual ;
                Get(CurrentDate)  ≥ window.contact::Call Next Connect Manual; "" ;
                window.contact::Call Next Connect Manual
                ) ]
        #
        Set Variable [ $NewItemType; Value:window.user::xNote_NoteType[ $pos ] ]
        Set Variable [ $$DragList; Value:window::xPt_ContactID ]
        Set Variable [ $$drag.portalID; Value:window::xPt_ContactID ]
        Set Variable [ $$drag.portal; Value:"external" ]
        Set Variable [ $$drag.note; Value:Trim( window.user::xNote_NoteNote[ $pos ] & " " & window::xSys_NewNote ) ]
        #
        #Set Primary Contact
```

```
    Set Field [ window::xPt_ItemIDForScript; $$drag.portalid ]
    Set Variable [ $PC; Value:Evaluate( join::xxIDL_FNLinkPrimaryItem ) ]
    #
    Set Field [ window::xPt_ItemIDForScript; Let( $link =
        Case(
        $NewItemType = "Email Received" ; "M";
        $NewItemType = "Email" ; "M";
        $NewItemType = "Communication" ; "Y";
        $NewItemType = "Note" ; "N";
        $NewItemType = "Missed" ; "X";
        $NewItemType = "VM" ; "V";
        $NewItemType = "Reg. History" ; "V";
        $NewItemType = "Print" ; "Z" ;
        "N"
        )   &
        Right("00" & Day(Get(CurrentDate));2)  &
        Right("00" & Month(Get(CurrentDate));2) &
        Right(Year(Get(CurrentDate));2)  &
        Right("00000" & GetAsText(Get(RecordID));5) &
        Right("0000" & GetAsText(Int(9999*Random));4)
        ; $link ) ]
    Set Field [ join::xxIDL_IDNumberPortal; $link ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Go to Related Record [ From table: “join”; Using layout: “data.join” (join) ]
        [ Show only related records; New window ]
    Set Field By Name [ join::xxIDL_FNIDNumberInternal; $link ]
    Set Field By Name [  join::xxIDL_FNLinkPrimaryItem; $PC ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Close Window [ Current Window ]
    #
    #link
    Perform Script [ “LinkItems [$$drop.id[1], $$draglist[2] ] / NewLinks [ $$drop.id [1] ]”; Parameter: Let( [$$drop.id = $link; $$draglist  =
        $$draglist ] ; "" ) ]
    Perform Script [ “Move/Copy/ClearToAnyFolder [ $$drop.ID [1], $$DragList [2] cleans, g(sp) = from folders ] ”; Parameter: Let( [

        $$drop.ID =
        Case(
        not IsEmpty( $folder) ; $folder ;
        PatternCount( window.user::xPrefUser_CopyToInboxWhenCreated; join::link.type.detailed ) > 0 ; $$inboxID ;
        "" ) ;

        $$DragList =
        If( PatternCount( window.user::xPrefUser_CopyToInboxWhenCreated; join::link.type.detailed ) > 0 or not IsEmpty( $folder );
        $link ;
        "" )

         ] ; "") ]
    Set Field [ window::xPt_ItemIDForScript; $link ]
    Set Field [ item.note::Note; $$drag.note ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    #
    Set Field [ window::xSys_NewItem[12]; $link ]
    Set Field [ window::xPt_ItemIDForScript; $lItemp ]
    #
Else
    Show Custom Dialog [ Message: "Error"; Buttons: “OK”, “Cancel” ]
    Exit Script [  ]
End If
#
Loop
    Exit Loop If [ Let( $count = $count + 1 ; $count)   >  12 ]
    Set Field [ window::xSys_NewItem[$count]; "" ]
End Loop
```

New: NewNote [type]

```
Set Field [ window::xSys_NewNoteType;  "Communication" ]
Set Field [ window::xSys_NewNote; "" ]
Set Field [ window::xSys_NewReminderDate; "" ]
Set Field [ window::xSys_NewReminderNote; "" ]
#
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Exit Script [ Result: Let( [
    $$EnterNote = "" ;
    $$drop.note = "" ;
    $$drag.ID = "" ;
    $$drop.ID = "" ;
    $$drag.portal = "";
    $$drag.portalID = "";
    $$drag.note = "";
    $$dragheadername = "";
    $$drop.sidebarID = "" ;
    $$drag.sidebarID = ""  ;
    $$Drag.foldertype = "" ;
    $$Draglist = "" ;
    $$Dontchange = ""
    ] ; "" ) ]
Show Custom Dialog [ Message: $pos  & "¶ type   " & window::xSys_NewNoteType & "¶ note   " & window::xSys_NewNote & "¶ pref
    note   " & window.user::xNote_NoteNote & "¶ nn24   " & window.user::xNote_NoteNote[24] & "¶ nn   " & window.user::
    xNote_NoteNote[$pos]; Buttons: “OK”, “Cancel” ]
```

New: NewNote2

```
Freeze Window
#[1] TYPE, [2] ITEMS TO LINK TO $$DragList, [3] PRIMARY ID $$drag.portalID, [4] PORTAL NAME (OR EXTERNAL) $$drag.portal, [5]
    NOTE or $$drag.note, [6] Reference ID
Set Variable [ $NewItemType; Value:Let( [
    $lltemp = window::xPt_ItemIDForScript ;
    $NoteType = Trim( window::xSys_NewItem[1] )
     ];
    Case(
    $NoteType = "Not In" or $NoteType = "Busy"; "New Missed" ;
    $NoteType = "Left VM" ; "New VM" ;

    $NoteType = "Communication" or $NoteType = "Busy" or $NoteType = "Busy, said to call back" or $NoteType = "Don't Call" or
    $NoteType = "Reffered to"  or $NoteType = "Gone, referred to"  or $NoteType= "Gone" or $NoteType = "Different VM" ; "New
    Communication" ;

    $NoteType = "Note" or $NoteType = "Wrong Number" or $NoteType= "Non Working Number" or $NoteType = "No Answer"  ; "New
    Note"  ;

    $NoteType
    )
    ) ]
Set Variable [ $$DragList; Value: window::xSys_NewItem[2] ]
Set Variable [ $$drag.portalID; Value:window::xSys_NewItem[3] ]
Set Variable [ $$drag.portal; Value: window::xSys_NewItem[4] ]
Set Variable [ $$drag.note; Value:window::xSys_NewItem[5] ]
Set Variable [ $passID; Value:window::xSys_NewItem[6] ]
#
#Set Primary Contact
Set Field [ window::xPt_ItemIDForScript; $$drag.portalid ]
Set Variable [ $PC; Value:Evaluate( join::xxIDL_FNLinkPrimaryItem ) ]
#
Set Field [ window::xPt_ItemIDForScript; Let( $link =
    Case(
    $NewItemType = "New Email Received" ; "M";
    $NewItemType = "New Email" ; "M";
    $NewItemType = "New Communication" ; "Y";
    $NewItemType = "New Note" ; "N";
    $NewItemType = "New Missed" ; "X";
    $NewItemType = "New VM" ; "V";
    $NewItemType = "New Reg. History" ; "V";
    $NewItemType = "New Print" ; "Z" ;
    "N"
    )   &
    Right("00" & Day(Get(CurrentDate));2)  &
    Right("00" & Month(Get(CurrentDate));2) &
    Right(Year(Get(CurrentDate));2)  &
    Right("00000" & GetAsText(Get(RecordID));5) &
    Right("0000" & GetAsText(Int(9999*Random));4)
    ; $link ) ]
Set Field [ join::xxIDL_IDNumberPortal; $link ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Go to Related Record [ From table: “join”; Using layout: “data.join” (join) ]
    [ Show only related records; New window ]
Set Field By Name [ join::xxIDL_FNIDNumberInternal; $link ]
Set Field By Name [  join::xxIDL_FNLinkPrimaryItem; $PC ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Close Window [ Current Window ]
#
#link
Perform Script [ “LinkItems [$$drop.id[1], $$draglist[2] ] / NewLinks [ $$drop.id [1] ]”; Parameter: Let( [$$drop.id = $link; $$draglist  = $
    $draglist ] ; "" ) ]
Perform Script [ “Move/Copy/ClearToAnyFolder [ $$drop.ID [1], $$DragList [2] cleans, g(sp) = from folders ] ”; Parameter: Let( [

    $$drop.ID =
    Case(
    not IsEmpty( $folder) ; $folder ;
```

```
        PatternCount( window.user::xPrefUser_CopyToInboxWhenCreated; join::link.type.detailed ) > 0 ; $$inboxID ;
        "" ) ;

        $$DragList =
        If( PatternCount( window.user::xPrefUser_CopyToInboxWhenCreated; join::link.type.detailed ) > 0 or not IsEmpty( $folder );$link ;
        "" )

         ] ; "") ]
Set Field [ window::xPt_ItemIDForScript; $link ]
Set Field [ item.note::Note; $$drag.note ]
Commit Records/Requests
        [ Skip data entry validation; No dialog ]
#
Set Field [ window::xSys_NewItem[12]; $link ]
Set Field [ window::xPt_ItemIDForScript; $lltemp ]
Exit Script [ Result: Let( [
        $$EnterNote = "" ;
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.portalid = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$Dontchange = ""
        ] ; "" ) ]
Show Custom Dialog [ Message: "1 "  & window::xSys_NewItem[ 1 ]  & "¶" &
        "2 "  & window::xSys_NewItem[ 2 ]  & "¶" &
        "3 "  & window::xSys_NewItem[ 3 ]  & "¶" &
        "4 "  & window::xSys_NewItem[ 4 ]  & "¶" &
        "5 "  & window::xSys_NewItem[ 5 ]  & "¶" &
        "6 "  & window::xSys_NewItem[ 6 ]  & "¶" &
        "1 "  & $sp  & "¶" &
        "2 "  & $$draglist  & "¶" &
        "3 "  & $$drag.portalid  & "¶" &
        "4 "  & $$drag.portal  & "¶" &
        "5 "  & $$drag.note  & "¶"; Buttons: “OK”, “Cancel” ]
```

New: NewContactSameCompany [$ac (old), link (new)]

```
Freeze Window
Allow User Abort [ Off ]
Set Variable [ $ac; Value:MiddleWords( Get(ScriptParameter) ; 1; 1) ]
Set Variable [ $link; Value:MiddleWords( Get(ScriptParameter) ; 2; 1) ]
If [ not IsEmpty(  $link ) ]
    New Window [ Name: "~temp~"; Height: 1; Width: 1; Top: -200 ]
    Freeze Window
    Go to Layout [ “data.window” (window) ]
    New Record/Request
    Set Field [ window::xPt_ContactID; $ac ]
    Set Field [ window::xPt_ItemIDForScript; $link ]
    Set Field [ item.contact::Company Name; window.contact::Company Name ]
    Set Field [ item.contact::Image; window.contact::Image ]
    Set Field [ item.contact::Region Overide; window.contact::Region Overide ]
    Set Field [ item.contact::Ship Address; window.contact::Ship Address ]
    Set Field [ item.contact::Ship City; window.contact::Ship City ]
    Set Field [ item.contact::Ship State; window.contact::Ship State ]
    Set Field [ item.contact::Ship Zip; window.contact::Ship Zip ]
    Set Field [ item.contact::Phone Area Code; window.contact::Phone Area Code ]
    Set Field [ item.contact::Phone; Left( window.contact::Phone ; 4) ]
    Set Field [ item.contact::Call Referral to; "From: " & window.contact::Name ]
    Set Field [ item.contact::Web Address; window.contact::Web Address ]
    If [ item.contact::Phone2 Description = "main" ]
        Set Field [ item.contact::Phone2; window.contact::Phone2 ]
        Set Field [ item.contact::Phone2 Area Code; window.contact::Phone2 Area Code ]
        Set Field [ item.contact::Phone2 Description; window.contact::Phone2 Description ]
    End If
    Close Window [ Current Window ]
End If
```

New: SameContactNewCompany [$ac (old), link (new)]

```
Freeze Window
Allow User Abort [ Off ]
Set Variable [ $ac; Value:MiddleWords( Get(ScriptParameter) ; 1; 1) ]
Set Variable [ $link; Value:MiddleWords( Get(ScriptParameter) ; 2; 1) ]
If [ not IsEmpty(  $link ) ]
    New Window [ Name: "~temp~"; Height: 1; Width: 1; Top: -350 ]
    Freeze Window
    Go to Layout [ “data.window” (window) ]
    New Record/Request
    Set Field [ window::xPt_ContactID; $ac ]
    Set Field [ window::xPt_ItemIDForScript; $link ]
    Set Field [ item.contact::Region Overide; window.contact::Region Overide ]
    Set Field [ item.contact::Ship State; window.contact::Ship State ]
    Set Field [ item.contact::Name; window.contact::Name ]
    Set Field [ window::xSys_NewItem; $ac ]
    Set Field [ window::xSys_NewItem[2]; $link ]
    Perform Script [ “LinkItems [$$drop.id[1], $$draglist[2] ] / NewLinks [ $$drop.id [1] ]” ]
    Close Window [ Current Window ]
End If
```

New: ScrollToItemAndSelect [link, $$drag.portal]

```
Freeze Window
If [ IsEmpty( Get (ScriptParameter) ) ]
    Set Variable [ $link; Value:window::xPt_SelectedItems__CurrentClickedPortalID ]
    Set Variable [ $dptemp; Value:$$drag.portal ]
    Set Variable [ $$drag.portal; Value:window::xPt_SelectedItems__CurrentClickedPortal ]
Else
    Set Variable [ $link; Value:Get(ScriptParameter) ]
End If
#
Loop
    #if "L" link item attempt to open parent
    Set Variable [ $OpenedTemp; Value:Evaluate( "window::xPT_Opened_"  & $$drag.portal  ) ]
    If [ Left( $link; 1) = "L" ]
        Set Variable [ $LPx; Value:If(
            Left( $link ; 1 )   ≠   "L" ; $item ;
            Middle ( $link ; Position ( $link ; "LPx" ; 1; 1) + 3; 20)
             ) ]
        Exit Loop If [ Case(
            PatternCount( List( Evaluate( "portal." & $$drag.portal & "::xxIDL_IDNumberPortal" ) ) ; $LPx ) = 0; Let( $f = "failed"; 1);
            ""
            ) ]
        Set Field By Name [ "window::xPT_Opened_"  & $$drag.portal; Let(  $L = Evaluate( "window::xPT_Opened_"  & $$drag.
        portal  )   ;

            Substitute( $L ; [  $LPx ; "¶" ] ; ["¶¶" ; "¶" ] )  & "¶" &  $LPx

            ) ]
    End If
    #
    #find location
    Set Variable [ $r; Value:Let( [
        $portalID = $link;
        $list =List( Evaluate( "portal." & $$drag.portal & "::xxIDL_IDNumberPortal" ) ) ;

        $PR = Position (  Filter( Substitute(  Substitute( "TTT" &  $list  & "TTT"; "¶" ; "TTT¶TTT" ) ; "TTT"  & $PortalID  & "TTT" ; "#") ;
        "#¶"  ) ; "#" ;1; 1)
        ];

        ""
        ) ]
    Exit Loop If [ Case(
        $PR < 1 ; Let( $f = "failed"; 1);
        ""
        ) ]
    #
    #goto portal row
    Go to Object [ Object Name: If( $$drag.portal = "mainlist" ;
        Case(
        GetLayoutObjectAttribute("tab.mainlist 3";"isFrontTabPanel")  =  1 ; "portal."  & $$drag.portal & " 3" ;
        GetLayoutObjectAttribute("tab.mainlist 2";"isFrontTabPanel")  =  1 ; "portal."  & $$drag.portal & " 2" ;
        "portal."  & $$drag.portal & ""
        )
        ;
         "portal."  & $$drag.portal
        ) ]
    Go to Portal Row [ $PR ]
        [ Select; No dialog ]
    Exit Loop If [ Case(
        Evaluate( "portal."  & $$drag.portal  & "::xxIDL_IDNumberPortal" )  ≠  $link ;  Let( $f = "failed"; 1);
        ""
        ) ]
    #
    Set Field [ window::xPt_SelectedItems__LastClickedPortal; window::xPt_SelectedItems__CurrentClickedPortal ]
    Set Field [ window::xPt_SelectedItems__LastClickedPortalID; window::xPt_SelectedItems__CurrentClickedPortalID ]
    Set Field [ window::xPt_SelectedItems__CurrentClickedPortal; $$drag.portal ]
    Set Field [ window::xPt_SelectedItems__CurrentClickedPortalID; $link ]
    Set Field By Name [ window::xPt_SelectedItems__CurrentClickedPortalFN; window::xPt_SelectedItems__CurrentClickedPortalID ]
    Set Variable [ $$drag.portal; Value:If( IsEmpty( Get (ScriptParameter) ) ; $DPtemp ; $$drag.portal) ]
    Exit Loop If [ 1 ]
End Loop
Set Field By Name [ Case(
    $f = "failed" ; "window::xPT_Opened_"  & $$drag.portal ;
    ""); $OpenedTemp ]
Exit Script [ Result: $f ]
```

New: NewFolder [$$NewFolder or Menu]

```
Freeze Window
#$$NewFolder[1] type ("folder", "smart folder")
$$NewFolder[2] name
$$NewFolder[3] contacts ID's
$$NewFolder[4] select
$$NewFolder[5] returned folder ID, $ac
( $$NewFolder[6] users list, all users )
Set Variable [ $sp; Value:Case(
    not IsEmpty( $$NewFolder[1] ) ; $$NewFolder[1] ; //"Folder" or "Contacts", etc.
    Trim(Substitute ( window::xSys_NewMenuChoice ; "New "; ""))
    ) ]
Set Field [ window::xSys_NewMenuChoice; " " ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
#
If [ IsEmpty ( $sp ) or $sp = " " or $sp = "Smart Folder -" ]
Else If [ $sp = "Folder" ]
    Set Field [ item.folder::xSys_NewFolderName; Case(
        not IsEmpty( $$NewFolder[1] ); $$NewFolder[2] ;
        "New Folder"
        ) ]
    If [  IsEmpty( $$NewFolder[1] ) ]
        Show Custom Dialog [ Title: "New Folder"; Buttons: “OK”, “Cancel”; Input #1: item.folder::xSys_NewFolderName, "Folder
            Name..." ]
        If [ Get(LastMessageChoice) = 2 ]
            Commit Records/Requests
                [ Skip data entry validation; No dialog ]
            Exit Script [ Result: Let( [
                $$NewFolder[1] = "" ;  //type
                $$NewFolder[2] = "" ;  //name
                $$NewFolder[3] = "" ;  //contacts
                $$NewFolder[4] = "" ;  //select
                $$NewFolder[5] = $ac ;
                $$drop.note = "" ;
                $$drag.ID = "" ;
                $$drag.PortalID = "" ;
                $$drop.ID = "" ;
                $$drag.portal = "";
                $$drag.note = "";
                $$dragheadername = "";
                $$drop.sidebarID = "" ;
                $$drag.sidebarID = ""  ;
                $$Drag.foldertype = "" ;
                $$Draglist = "" ;
                $$DraglistPortalID = "" ;
                $$Dontchange = ""
                ] ; $ac ) ]
        End If
    End If
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    If [ Get ( LastMessageChoice ) = 1 ]
        New Window [ Name: "~temp~"; Height: 1; Width: 1; Top: -400 ]
        Go to Layout [ “data.folder” (item.folder) ]
        New Record/Request
        Set Field [ item.folder::Folder Name; item.folder::xSys_NewFolderName ]
        Set Field [ item.folder::xxIDL_Owner; Get ( AccountName ) ]
        Set Field [ item.folder::xxIDL_LinkList; Case(
            not IsEmpty( $$NewFolder[1] ); $$NewFolder[3] ;
            ""
            ) ]
        Set Variable [ $ac; Value:item.folder::xxIDL_IDNumberInternal ]
        Close Window [ Current Window ]
        Perform Script [ “LinkItems [$$drop.id[1], $$draglist[2] ] / NewLinks [ $$drop.id [1] ]”; Parameter: Let( [ $$drop.id = $ac ; $
            $draglist = "" ] ; "") ]
    End If
Else
```

```
Else
    Set Field [ item.folder::xSys_NewFolderName; Case(
        not IsEmpty( $$NewFolder[1] ); $$NewFolder[2] ;
        "New Folder"
        ) ]
    If [  IsEmpty( $$NewFolder[1] ) ]
        Show Custom Dialog [ Title: "New Smart Folder"; Buttons: “OK”, “Cancel”; Input #1: item.folder::xSys_NewFolderName, "Smart
            Folder Name (" & Trim($sp) & ")..." ]
        If [ Get(LastMessageChoice) = 2 ]
            Commit Records/Requests
                [ Skip data entry validation; No dialog ]
            Exit Script [ Result: Let( [
                $$NewFolder[1] = "" ;  //type
                $$NewFolder[2] = "" ;  //name
                $$NewFolder[3] = "" ;  //contacts
                $$NewFolder[4] = "" ;  //select
                $$NewFolder[5] = $ac ;
                $$drop.note = "" ;
                $$drag.ID = "" ;
                $$drag.PortalID = "" ;
                $$drop.ID = "" ;
                $$drag.portal = "";
                $$drag.note = "";
                $$dragheadername = "";
                $$drop.sidebarID = "" ;
                $$drag.sidebarID = ""  ;
                $$Drag.foldertype = "" ;
                $$Draglist = "" ;
                $$DraglistPortalID = "" ;
                $$Dontchange = ""
                ] ; $ac ) ]
        End If
    End If
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    If [ Get ( LastMessageChoice ) = 1 ]
        New Window [ Name: "~temp~"; Height: 1; Width: 1; Top: -400 ]
        Go to Layout [ “data.folder” (item.folder) ]
        New Record/Request
        Set Field [ item.folder::Folder Name; item.folder::xSys_NewFolderName ]
        Set Field [ item.folder::Folder SF type; Case(
            $sp = "Actions"; "Action" ;
            $sp = "Registration"; "Registration" ;
            $sp = "Quotes"; "Quote" ;
            $sp = "Orders"; "Order" ;
            $sp = "Equipment"; "Equipment" ;
            $sp = "Notes"; "Note" ;
            "Contact"
            ) ]
        Set Field [ item.folder::Folder Type; "Smart Folder" ]
        Set Field [ item.folder::xxIDL_Owner; Get ( AccountName ) ]
        Set Variable [ $ac; Value:item.folder::xxIDL_IDNumberInternal ]
        Close Window [ Current Window ]
        Perform Script [ “LinkItems [$$drop.id[1], $$draglist[2] ] / NewLinks [ $$drop.id [1] ]”; Parameter: Let( [ $$drop.id = $ac ; $
            $draglist = "" ] ; "") ]
    End If
End If
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
If [ IsEmpty( $$NewFolder[1] ) ]
    Exit Script [ Result: Let( [
        $$NewFolder[1] = "" ;  //type
        $$NewFolder[2] = "" ;  //name
        $$NewFolder[3] = "" ;  //contacts
        $$NewFolder[4] = "" ;  //select
        $$NewFolder[5] = $ac ;
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drag.PortalID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$DraglistPortalID = "" ;
        $$Dontchange = ""
        ] ; $ac ) ]
Else
    Exit Script [ Result: Let( [
        $$NewFolder[1] = "" ;  //type
        $$NewFolder[2] = "" ;  //name
        $$NewFolder[3] = "" ;  //contacts
        $$NewFolder[4] = "" ;  //select
        $$NewFolder[5] = $ac
        ] ; $ac ) ]
End If
```

New: email from AS

```
#--- do not change script name!!!
#property thescript : "email from AS"
property scripttype : "sender"
property scriptaction : "move goto"
property theSelection : ""
property theSelected : ""
property theMessage : ""
property thename : ""
property thesubject : ""
property thecontent : ""
property thedate : ""
property theresultmess : ""
property theRecipientAdd : ""
property theRecipientName : ""
property theheader : ""

tell application "Mail"
    set theSelection to selection
    set theMessage to item 1 of theSelection
    set theAddress to extract address from sender of theMessage
    set thename to extract name from sender of theMessage
    set thesubject to subject of theMessage
    set theheader to all headers of theMessage
    set thecontent to content of theMessage
    set {year:y, month:m, day:d} to date sent of theMessage
    set thedate to (m * 1 as string) & "/" & (d * 1 as string) & "/" & (y * 1 as string)
end tell

tell application "FileMaker Pro"
    tell database "customers.fp7"

        set prevTIDs to AppleScript's text item delimiters
        set AppleScript's text item delimiters to ASCII character 10
        set thecontent to text items of thecontent
        set AppleScript's text item delimiters to ASCII character 13
        set thecontent to thecontent as text
        set AppleScript's text item delimiters to prevTIDs

        set theSelected to "__OBJ_TEXT__"
        set prevTIDs to AppleScript's text item delimiters
        set AppleScript's text item delimiters to ASCII character 10
        set theSelected to text items of theSelected
        set AppleScript's text item delimiters to ASCII character 13
        set theSelected to theSelected as text
        set AppleScript's text item delimiters to prevTIDs

        set cellValue of repetition 1 of cell "sys.as.pass" of table "window" to theSelected
        set cellValue of repetition 2 of cell "sys.as.pass" of table "window" to theAddress
        set cellValue of repetition 3 of cell "sys.as.pass" of table "window" to thename
        set cellValue of repetition 4 of cell "sys.as.pass" of table "window" to thesubject
        set cellValue of repetition 5 of cell "sys.as.pass" of table "window" to thecontent
        set cellValue of repetition 6 of cell "sys.as.pass" of table "window" to thedate
        set cellValue of repetition 8 of cell "sys.as.pass" of table "window" to scripttype
        set cellValue of repetition 9 of cell "sys.as.pass" of table "window" to scriptaction
        set cellValue of repetition 10 of cell "sys.as.pass" of table "window" to ""
        set cellValue of repetition 11 of cell "sys.as.pass" of table "window" to theheader
        do script FileMaker script thescript
    end tell
end tell
```

```
Freeze Window
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
New Window [ Name: "~temp~"; Height: 1; Width: 1; Top: -350 ]
Freeze Window
#
Go to Layout [ “data.join” (join) ]
New Record/Request
Set Field [ join::xxIDL_IDNumberPortal; Let( $link =
    "M"  &
    Right("00" & Day(Get(CurrentDate));2)  &
    Right("00" & Month(Get(CurrentDate));2) &
    Right(Year(Get(CurrentDate));2)  &
    Right("00000" & GetAsText(Get(RecordID));5) &
    Right("0000" & GetAsText(Int(9999*Random));4)
    ; $link ) ]
```

```
Commit Records/Requests
        [ Skip data entry validation; No dialog ]
Set Field By Name [ Evaluate( join::xxIDL_FNIDNumberInternal); $link ]
Set Field [ item.note::xxIDL_IDNumberInternal; $link ]
Commit Records/Requests
        [ Skip data entry validation; No dialog ]
#
Go to Layout [ “data.window” (window) ]
New Record/Request
Set Field [ window::xPt_ItemIDForScript; $link ]
Set Variable [ $end; Value:Let(   [
        $LL = window::sys.as.pass[11] ;
        $count = 0
        ];
        ValueCount($LL)  ) ]
Loop
        Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
        Set Field [ item.note::Email Recipient List; Let( [
                $b = MiddleValues (window::sys.as.pass[11] ; $count ; 1 )
                ] ; If(  Left( $b ; 4) = "To: " ; Trim ( Substitute ( $b ; [ "To: " ; "" ] ; [ "¶" ; "" ] ;["<";""] ;[">";""] ; ["\"" ; "" ])) ; item.note::Email Recipient
                List )
                ) ]
        Set Field [ item.note::Email cc; Let( [
                $b = MiddleValues (window::sys.as.pass[11] ; $count ; 1 )
                ] ; If(  Left( $b ; 4) = "Cc: " ; Trim ( Substitute ( $b ; [ "Cc: " ; "" ] ; [ "¶" ; "" ] ;["<";""] ;[">";""] ; ["\"" ; "" ])) ; item.note::Email cc )
                ) ]
End Loop
Set Field [ item.note::Note; If( PatternCount( window::sys.as.pass[5] ; "¶" ) > 3 ;

        Let($c = Filter ( Substitute ( window::sys.as.pass[5] ; Char(8232) ; "¶" ); "~!@#$%^&*()_+`1234567890-=QWERTYUIOP{}|qwertyuiop
        []\ASDFGHJKL:asdfghjkl;'ZXCVBNM<>?zxcvbnm,./ ¶"  & "\"")  ;
        Substitute ( Trim ( Substitute ( $c ; [ " " ; "#@#@#@!" ] ; [ "¶" ; " "] ) ) ; [ " " ; "¶" ] ; [ "#@#@#@!"; " "] ;
         [ Char(10) ; "¶" ] ;["¶¶¶"; "¶¶"] ;  ["¶¶¶"; "¶¶"] ; ["¶¶¶"; "¶¶"]  ; ["¶¶¶"; "¶¶"] ; ["¶¶¶"; "¶¶"]  ; ["¶¶¶"; "¶¶"] )
        ) ;
        window::sys.as.pass[5]
        ) ]
Set Field [ item.note::Email Sender Email; window::sys.as.pass[2] ]
Set Field [ item.note::Email Sender Name; window::sys.as.pass[3] ]
Set Field [ item.note::Email Subject; window::sys.as.pass[4] ]
Set Field [ item.note::Date; GetAsDate(window::sys.as.pass[6]) ]
Set Field [ item.note::Email Status; "Imported" ]
Set Variable [ $MyEmail; Value:window.user::user.email ]
#
Set Field [ window::xID_EmailimportKey; Substitute(
        item.note::Email Sender Email
        ; [ $MyEmail ; "¶" ] ;[" ";"¶"];[":";"¶"];[";";"¶"];[",";"¶"] ;["(";"¶"];[")";"¶"] ;["?";"¶"] ;["\"";"¶"];["<";"¶"] ;[">";"¶"] ) ]
Set Field [ item.note::xxIDL_LinkPrimaryItem; Case(
        IsEmpty(item.note::xxIDL_LinkPrimaryItem ) ; LeftWords( window.emaillookup.contactemail::xxIDL_IDNumberInternal ; 1 );
        item.note::xxIDL_LinkPrimaryItem
        ) ]
If [ not IsEmpty( item.note::xxIDL_LinkPrimaryItem ) ]
        Set Field [ window::xPt_ContactID; item.note::xxIDL_LinkPrimaryItem ]
        Set Field [ window.contact::Call Last Connect; Get(CurrentDate) ]
End If
Set Field [ window::xID_EmailimportKey; Substitute(
        item.note::Email cc
        ; [ $MyEmail ; "¶" ] ;[" ";"¶"];[":";"¶"];[";";"¶"];[",";"¶"] ;["(";"¶"];[")";"¶"] ;["?";"¶"] ;["\"";"¶"];["<";"¶"] ;[">";"¶"] ) ]
Set Field [ item.note::xxIDL_LinkPrimaryItem; Case(
        IsEmpty(item.note::xxIDL_LinkPrimaryItem ) ; LeftWords( window.emaillookup.contactemail::xxIDL_IDNumberInternal ; 1 );
        item.note::xxIDL_LinkPrimaryItem
        ) ]
Set Field [ window::xID_EmailimportKey; Substitute(
        item.note::Email Recipient List
        ; [ $MyEmail ; "¶" ] ;[" ";"¶"];[":";"¶"];[";";"¶"];[",";"¶"] ;["(";"¶"];[")";"¶"] ;["?";"¶"] ;["\"";"¶"];["<";"¶"] ;[">";"¶"] ) ]
Set Field [ item.note::xxIDL_LinkPrimaryItem; Case(
        IsEmpty(item.note::xxIDL_LinkPrimaryItem ) ; LeftWords( window.emaillookup.contactemail::xxIDL_IDNumberInternal ; 1 );
        item.note::xxIDL_LinkPrimaryItem
        ) ]
#
Set Field [ window::xID_EmailimportKey; Substitute(  item.note::Email Recipient List  & " " & item.note::Email Sender Email  & " " & item.
        note::Email cc  & " "  & item.note::Note ; [ $MyEmail ; "¶" ] ;[" ";"¶"];[":";"¶"];[";";"¶"];[",";"¶"] ;["(";"¶"];[")";"¶"] ;["?";"¶"] ;["\"";"¶"];["<";"¶"] ;
        [">";"¶"] ) ]
Commit Records/Requests
        [ Skip data entry validation; No dialog ]
Perform Script [ “LinkItems [$$drop.id[1], $$draglist[2] ] / NewLinks [ $$drop.id [1] ]”; Parameter: Let( [
        $$drop.id = $link ;
        $$draglist =
        If( not IsEmpty( window.emaillookup.contactemail::xxIDL_IDNumberInternal  & window.emaillookup.registration::
        xxIDL_IDNumberInternal  )  ;
        List( window.emaillookup.contactemail::xxIDL_IDNumberInternal )  & "¶" & List( window.emaillookup.registration::
        xxIDL_IDNumberInternal );
        "" )

        ] ; "" ) ]
Perform Script [ “Move/Copy/ClearToAnyFolder [ $$drop.ID [1], $$DragList [2] cleans, g(sp) = from folders ] ”; Parameter: Let( [ $$drop.id
        = $$inboxID ; $$draglist = $link ] ;"") ]
If [ item.note::xxIDL_LinkPrimaryItem ]
End If
#
Close Window [ Current Window ]
Set Field [ window::xID_ReminderID; "" ]
Perform Script [ “ImportEmailWindow [ link ]”; Parameter: $link ]
Set Field [ window::xPt_ItemIDForScript; "" ]
Perform AppleScript [ Native AppleScript: tell application "Mail"
              activate
        end tell
         ]
Exit Script [ Result: Let( [
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drag.PortalID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$DraglistPortalID = "" ;
        $$Dontchange = ""
        ] ; "" ) ]
```

New: UserNote [new, previous, next, newest, oldest, delete, select, style]

```
Freeze Window
Set Variable [ $sp; Value:Get(ScriptParameter) ]
Set Variable [ $pos; Value:Let( [
    $min = Min ( UserNotesAll::xID_UserNoteID );
    $max = Max ( UserNotesAll::xID_UserNoteID ) ;
    $item = window::xID_UserNote ;
    $list = GetAsText( List( UserNotesAll::xID_UserNoteID  )  ) ;
    $PR = Position (  Filter( Substitute(  Substitute( "TTT" &  $list  & "TTT"; "¶" ; "TTT¶TTT" ) ; "TTT"  & $item  & "TTT" ; "#") ; "#¶"  ) ;
    "#" ;1; 1) ;
    $current = MiddleWords( $list; $PR; 1)

    ];
    $PR
    ) ]
If [ $sp = "New" ]
    Set Field [ window::xID_UserNote; $max + 1 ]
    Set Field [ UserNotes::xID_UserNoteName; "Note " & $max + 1 ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
Else If [ $sp = "delete" ]
    If [ Count( UserNotesAll::xID_UserNoteFlag ) > 1 ]
        Show Custom Dialog [ Message: "Delete this note?"; Buttons: “OK”, “Cancel” ]
        If [ Get ( LastMessageChoice )  = 1 ]
            Set Field [ UserNotes::xxIDL_IDNumberInternal; "delete" ]
            Set Field [ UserNotes::xID_UserNoteID; "delete" ]
            Commit Records/Requests
                [ Skip data entry validation; No dialog ]
            Perform Script [ “UserNote [new, previous, next, newest, oldest, delete, select, style]”; Parameter: "previous" ]
        End If
    End If
Else If [ $sp = "select" ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
Else If [ $sp = "next" ]
    Set Field [ window::xID_UserNote; Case(
        $max = $current; $max ;
        GetAsNumber( MiddleWords ( $list ; $pos - 1 ; 1 ) )
        ) ]
Else If [ $sp = "previous" ]
    Set Field [ window::xID_UserNote; Case(
        $min = $current; $min ;
        GetAsNumber( MiddleWords ( $list ; $pos + 1 ; 1 ) )
        ) ]
Else If [ $sp = "newest" ]
    Set Field [ window::xID_UserNote; $max ]
Else If [ $sp = "oldest" ]
    Set Field [ window::xID_UserNote; $min ]
Else If [ $sp = "style" ]
    If [ Get ( ActiveFieldName ) = "Note" and
        Get ( ActiveFieldTableName ) = "UserNotes" and
        Get ( ActiveSelectionSize ) > 0 ]
        Show Custom Dialog [ Message: "Remove font type & size or remove all (type, size, color, styles) for the selected text?";
            Buttons: “Type & Size”, “All”, “Cancel” ]
        Set Variable [ $start; Value:Get ( ActiveSelectionStart ) ]
        Set Variable [ $size; Value:Get ( ActiveSelectionSize ) ]
        If [ Get ( LastMessageChoice ) = 1 ]
            Set Field [ UserNotes::Note; Middle ( UserNotes::Note  ; 1 ; Get ( ActiveSelectionStart ) - 1 )  & TextFontRemove
            ( TextSizeRemove ( Middle ( UserNotes::Note  ;  Get ( ActiveSelectionStart ) ; Get ( ActiveSelectionSize )) ))  & Middle
            ( UserNotes::Note  ;  Get ( ActiveSelectionStart ) + Get ( ActiveSelectionSize ) ; Length (UserNotes::Note)  )

                //Middle ( text ; start ; numberOfCharacters ) ]
        Else If [ Get ( LastMessageChoice ) = 2 ]
            Set Field [ UserNotes::Note; Middle ( UserNotes::Note  ; 1 ; Get ( ActiveSelectionStart ) - 1 )  & TextFormatRemove
            ( Middle ( UserNotes::Note  ;  Get ( ActiveSelectionStart ) ; Get ( ActiveSelectionSize )) )  & Middle ( UserNotes::Note  ;
            Get ( ActiveSelectionStart ) + Get ( ActiveSelectionSize ) ; Length (UserNotes::Note)  )

                //Middle ( text ; start ; numberOfCharacters ) ]
        End If
        Set Selection [ UserNotes::Note; Start Position: $start; End Position: $start+ $size - 1 ]
        Exit Script [  ]
    Else
        Show Custom Dialog [ Message: "Remove font type & size or remove all (type, size, color, styles) for this note?"; Buttons:
            “Type & Size”, “All”, “Cancel” ]
        If [ Get ( LastMessageChoice ) = 1 ]
            Set Field [ UserNotes::Note; TextFontRemove ( TextSizeRemove ( UserNotes::Note  ) ) ]
        Else If [ Get ( LastMessageChoice ) = 2 ]
            Set Field [ UserNotes::Note; TextFormatRemove (  UserNotes::Note  ) ]
        End If
    End If
End If
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
```

New: NotePicture

```
Set Field [ window::xSys_NotePictureEdit; Case(
        Get(ScriptParameter) = "enter"; "✎";
        "<"
        ) ]
```

Window: WindowNew [main, folder, folder options, pref, note, newemail, importemail, phonerecord…

```
Freeze Window
#Open Window
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
If [ Get(ScriptParameter) = "Folder options" ]
    New Window [ Height: 518; Width: 797; Top: 301; Left: 408 ]
    Go to Layout [ “Folder Options” (window) ]
    Freeze Window
Else If [ Get(ScriptParameter) = "newemail" ]
    New Window [ Name: "Email"; Height: 680; Width: 720; Top: 52; Left: 350 ]
    Go to Layout [ “New Email Message” (window) ]
Else If [ Get(ScriptParameter) = "websites" ]
    New Window [ Height: 657; Width: 700; Top: 4; Left: 40 ]
    Go to Layout [ “Websites” (window) ]
Else If [ Get(ScriptParameter) = "importemail" ]
    If [ PatternCount ( WindowNames  ; "Import Email" )  = 0 ]
        New Window [ Name: "Import Email"; Height: 822; Width: 806; Top: 9; Left: 28 ]
        Go to Layout [ “Import Email” (window) ]
    Else
        Select Window [ Name: "Import Email"; Current file ]
        Exit Script [  ]
    End If
Else If [ Get(ScriptParameter) = "phonerecord" ]
    If [ PatternCount ( WindowNames  ; "Phone Record" )  = 0 ]
        New Window [ Name: "Phone Record"; Height: 77; Width: 288; Top: 0; Left: 1185 ]
        Go to Layout [ “Phone Record” (window) ]
        Show/Hide Status Area
            [ Hide ]
        Move/Resize Window [ Current Window; Height: 77; Width: 288; Top: 0; Left: 1185 ]
    Else
        Select Window [ Name: "Phone Record"; Current file ]
        Exit Script [  ]
    End If
Else If [ Get(ScriptParameter) = "note" ]
    If [ Get ( WindowWidth ) < 1210 ]
        Move/Resize Window [ Current Window; Width: 1424 ]
    Else
        Move/Resize Window [ Current Window; Width: 1190 ]
        Scroll Window
            [ Home ]
    End If
    Exit Script [ Result: Let( [
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drag.PortalID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$DraglistPortalID = "" ;
        $$Dontchange = ""
        ] ; "" ) ]
Else If [ Get(ScriptParameter) = "pref" ]
    New Window [ Name: "Preferences"; Height: 550; Width: 765; Top: 151; Left: 274 ]
    Go to Layout [ “Preferences” (window.contact.user) ]
    Freeze Window
    Exit Script [  ]
    Exit Script [ Result: Let( [
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drag.PortalID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$DraglistPortalID = "" ;
        $$Dontchange = ""
        ] ; "" ) ]
Else If [ Get(ScriptParameter) = "main" ]
    New Window [ Top: Get ( WindowTop ) +15; Left: Get ( WindowLeft ) +15 ]
    Install OnTimer Script [ “Timer”; Interval: 300 ]
End If
#
#New window ID number
If [ Get(ScriptParameter) = "main" or Get(ScriptParameter) = "Folder options" ]
    Duplicate Record/Request
    Set Field [ window::xID_SpotlightEntry; "" ]
    Set Field [ window::xxIDL_IDNumberInternal; "W" &
        Right("00" & Day(Get(CurrentDate));2)  &
        Right("00" & Month(Get(CurrentDate));2) &
        Right(Year(Get(CurrentDate));2)  &
        Right("00000" & GetAsText(Get(RecordID));5) &
        Right("0000" & GetAsText(Int(9999*Random));4) ]
Else
    New Record/Request
End If
#
#Item Tab
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Perform Script [ “ItemTab” ]
#
#Goto Fields
If [ Get(ScriptParameter) = "Folder" ]
Else If [ Get(ScriptParameter) = "websites" ]
    Set Field [ window::xID_UserWebType; "Customers" ]
Else If [ Get(ScriptParameter) = "Folder options" ]
    Set Field [ window::xID_FolderHeaderTypeExclude; "LISTS¶USERS" ]
Else If [ Get(ScriptParameter) = "note" ]
    Go to Field [ window.user::user.my notes ]
    Insert Calculated Result [ Case(
        IsEmpty( window.user::user.my notes[ 1 ] ) ; "" ;
        Right( window.user::user.my notes[ 1 ] ; 2 ) = "¶¶" ; "" ;
        "¶¶"
        ) ]
Else
End If
Exit Script [  ]
```

Window: NewItemGotoField [1, 2...]

Go to Field [window::xSys_NewNote[Get(ScriptParameter)]]

Window: WindowOpenRegisterRenew

```
Set Error Capture [ On ]
Freeze Window
Commit Records/Requests
        [ Skip data entry validation; No dialog ]
Set Variable [ $a; Value:Let( [
        $ac = window.contact::xxIDL_IDNumberInternal;
        $link = window::xPt_ItemID;
        $sp = Get(ScriptParameter);
        $mk = Get ( ActiveModifierKeys )
        ];"")

        /*
        Shift = 1,
        Caps Lock = 2,
        Ctrl (Windows) and Control (Mac OS) = 4,
        Alt (Windows) and Option (Mac OS) = 8,
        ⌘ (Mac OS) = 16
        */ ]
New Window [ Name: "Register/Renew" ]
Go to Layout [ “Register/Renew” (window) ]
New Record/Request
Perform Script [ “WindowResize [centered, full, ]” ]
Set Field [ window::xPt_ItemID; $Link ]
Commit Records/Requests
        [ Skip data entry validation; No dialog ]
Exit Script [ Result: Let( [
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drag.PortalID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$DraglistPortalID = "" ;
        $$Dontchange = ""
        ] ; "" ) ]
```

Window: WindowResize [centered, full,]

```
Set Variable [ $sp; Value:If(Get(LayoutName )= "Contact card"; "full" ; Get(ScriptParameter)) ]
Freeze Window
Show/Hide Status Area
        [ Hide ]
Show/Hide Text Ruler
        [ Hide ]
Set Zoom Level
        [ 100% ]
Adjust Window
        [ Resize to Fit ]
Move/Resize Window [ Current Window; Height: If($sp = "full"; Get ( WindowDesktopHeight ) - 120 ;
        "" ); Width: If(Get ( LayoutName ) = "Contact Card"; Get ( WindowDesktopWidth ) -  280; ""); Top: If(Get(ScriptParameter )=
        "centered" ; ((Get ( WindowDesktopHeight ) - Get ( WindowContentHeight )) / 2 )  ;

        Let( $n = PatternCount(WindowNames ; "¶" );
        Case(
        IsEmpty(WindowNames) ; 60;
        $n < 8   ; 60 + (PatternCount(WindowNames ; "¶" ) ) * 15  ;
        $n < 16 ; 60 + (PatternCount(WindowNames ; "¶" )  -8 ) * 15  ;
        60 + (PatternCount(WindowNames ; "¶" )  -16 ) * 15
        ))); Left: If(Get(ScriptParameter )= "centered" ; ((Get ( WindowDesktopWidth ) - Get ( WindowWidth ) ) / 2  )  - 160;

        30 + If(IsEmpty(WindowNames) ; "";    (PatternCount(WindowNames ; "¶" ) ) * 20  )

        ) ]
```

Window: WindowShowSize

```
Show Custom Dialog [ Message: "height: " & Get ( WindowHeight ) & "¶" &
        "width: " & Get ( WindowWidth ) & "¶" &
        "top: " & Get ( WindowTop ) & "¶" &
        "left: " & Get ( WindowLeft ) & "¶¶" &
        WindowNames  & "hhhhhhh"; Buttons: “OK”, “Cancel” ]
```

Window: Save Window Size for Main Window

```
Set Field [ window.user::xPrefUser_MainWindowSize; Get ( WindowHeight )

        /*
        "1height: " & Get ( WindowHeight ) & "¶" &
        "2width: " & Get ( WindowWidth ) & "¶" &
        "3top: " & Get ( WindowTop ) & "¶" &
        "4left: " & Get ( WindowLeft )
        */ ]
Set Field [ window.user::xPrefUser_MainWindowSize[2]; Get ( WindowWidth )

        /*
        "1height: " & Get ( WindowHeight ) & "¶" &
        "2width: " & Get ( WindowWidth ) & "¶" &
        "3top: " & Get ( WindowTop ) & "¶" &
        "4left: " & Get ( WindowLeft )
        */ ]
Set Field [ window.user::xPrefUser_MainWindowSize[3];  Get ( WindowTop )

        /*
        "1height: " & Get ( WindowHeight ) & "¶" &
        "2width: " & Get ( WindowWidth ) & "¶" &
        "3top: " & Get ( WindowTop ) & "¶" &
        "4left: " & Get ( WindowLeft )
        */ ]
Set Field [ window.user::xPrefUser_MainWindowSize[4]; Get ( WindowLeft )

        /*
        "1height: " & Get ( WindowHeight ) & "¶" &
        "2width: " & Get ( WindowWidth ) & "¶" &
        "3top: " & Get ( WindowTop ) & "¶" &
        "4left: " & Get ( WindowLeft )
        */ ]
```

Window: ImportEmailWindow [link]

```
Set Variable [ $link; Value:LeftWords( Get(ScriptParameter) ; 1) ]
Perform Script [ “WindowNew [ main, folder, folder options, pref, note, newemail, importemail, phonerecord, websites]”; Parameter:
    "importemail" ]
If [ window::xPt_SelectedItems__CurrentClickedPortalID  ≠ $link ]
    Set Field [ window::xPt_SelectedItems__CurrentClickedPortal; "MainList" ]
    Set Field [ window::xPt_SelectedItems__CurrentClickedPortalID; $link ]
    Set Field [ window::xPt_SelectedItems_MainList; $link ]
End If
#
Set Field [ window::xPt_ItemID; $link ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Set Field [ window::xID_EmailimportKey; Substitute( item.note::Email Sender Email & " "  &  item.note::Email Sender Name  & " " & item.
    note::Email Recipient List  & " " & item.note::Email cc  & " "  & item.note::Note ;[" ";"¶"];[":";"¶"];[";";"¶"];[",";"¶"] ;["(";"¶"];[")";"¶"] ;
    ["?";"¶"] ;["\"";"¶"];["<";"¶"] ;[">";"¶"] ) ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
If [ not IsEmpty( item.note::xxIDL_LinkPrimaryItem ) and  Left(LeftWords(item.note::xxIDL_LinkPrimaryItem ;1) ; 1) = "c" ]
    Set Field [ window::xPt_ContactID; item.note::xxIDL_LinkPrimaryItem ]
End If
#
Set Variable [ $end; Value:Let(   [
    $LL = join::xxIDL_LinkListTotal ;
    $count = 1  ;
    $Iltemp = window::xPt_ItemIDForScript
    ];
    WordCount($LL)  ) ]
Loop
    Exit Loop If [ $count  >  $end ]
    Set Variable [ $link; Value: MiddleWords($LL; $count ; 1 ) ]
    Set Variable [ $reg; Value:If( Left( $link; 1) = "R" ; $link ; $reg) ]
    Set Variable [ $count; Value:$count + 1 ]
End Loop
Set Field [ window::xID_EmailimportRegistration; $reg ]
#
Set Field [ window::xID_LinkItemsItemID; $link ]
Set Field [ window::xPt_MainListLayout; "importemail" ]
Set Field [ window::xPt_ItemIDForScript; "" ]
Exit Script [  ]
```

Window: NoteButtons [enter, exit, edit, connectdate]

```
If [ LeftWords( Get(ScriptParameter) ; 1) = "enter" or LeftWords( Get(ScriptParameter) ; 1) = "exit" ]
    Set Field [ window::xSys_NotePictureEdit; Case(
        RightWords( Get(ScriptParameter) ;1 )  = "enter"; "✎";
        ""
        ) ]
    Set Field [ window::xSys_NotePictureConnectDate; Case(
        RightWords( Get(ScriptParameter) ;1 )  = "enter"; "☼";
        ""
        ) ]
    Set Field [ window::xSys_NotePictureDelete; Case(
        RightWords( Get(ScriptParameter) ;1 )  = "enter"; "X";
        ""
        ) ]
    Set Selection [ window::xSys_ContactNotes; Start Position: Case(
        Length( window::xSys_ContactNotes ) < 2 ; 1;
        Get ( ActiveSelectionStart )  ≥ Length( window::xSys_ContactNotes ) ; 1 ;
        Get ( ActiveSelectionStart )
        ); End Position: Case(
        Get ( ActiveSelectionSize ) > 0 ; Get ( ActiveSelectionStart ) + Get ( ActiveSelectionSize ) - 1 ;
        ""
        ) ]
    Exit Script [  ]
End If
Set Variable [ $notes; Value:Substitute (
    List(  portal.notes::link.notes  ) ;
    [  "#dot#" & Char( 9 )  &  window.contact::xSys_LastNameFirstInitial   & "  " ; "#dot#" & Char( 9 )  ] ;
    [  "#dot#" & Char( 9 ) ;  Char( 9 ) ] ;
    [ "#start#" ; "•" ]
    ) ]
If [ LeftWords( Get(ScriptParameter) ; 1) = "edit" and
    Get ( ActiveFieldName ) = "xSys_ContactNotes"

    //and Get ( ActiveSelectionSize ) > 0 ]
    Set Variable [ $pos; Value:Let( [
        $start = Get ( ActiveSelectionStart ) ;
        $length = Get ( ActiveSelectionSize ) ;
        $item = "Å" ;
        $list =   Substitute ( Filter( Replace( $notes ; $start;  $length ;  $item) ; "•Å" ) ; "•" ; "¶" ) ;
        $PR = Position (  Filter( Substitute(  Substitute( "TTT" &  $list  & "TTT"; "¶" ; "TTT¶TTT" ) ; "TTT"  & $item  & "TTT" ; "#") ;
        "#¶"  ) ; "#" ;1; 1) ;
        $link = MiddleWords( List( portal.notes::link.notesID) ; $PR - 1 ; 1 );
        $$drag.portal = "notes"
        ];
        $PR - 1
        )

        /*
        $NewStart = $start - Position ( window::xSys_ContactNotes ; "•" ; 1 ; $PR - 1 ) - 2 ;
        $NewLength = $length
        */ ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Perform Script [ “DoubleClicked [ link (optional), $$EnterNote , $$drag.portal]”; Parameter: $link ]
    Exit Script [  ]
    Go to Object [ Object Name: "field.note.note" ]
    Set Selection [ Start Position: If( $newStart < 1; 1; $newstart); End Position:  $newstart +

        $NewLength ]
Else If [ LeftWords( Get(ScriptParameter) ; 1) = "connectdate" and
    Get ( ActiveFieldName ) = "xSys_ContactNotes"
    //and Get ( ActiveSelectionSize ) > 0 ]
    Set Variable [ $pos; Value:Let( [
        $start = Get ( ActiveSelectionStart ) ;
        $length = Get ( ActiveSelectionSize ) ;
        $item = "Å" ;
        $list =   Substitute ( Filter( Replace( $notes ; $start;  $length ;  $item) ; "•Å" ) ; "•" ; "¶" ) ;
        $PR = Position (  Filter( Substitute(  Substitute( "TTT" &  $list  & "TTT"; "¶" ; "TTT¶TTT" ) ; "TTT"  & $item  & "TTT" ; "#") ;
        "#¶"  ) ; "#" ;1; 1) ;
        $link = MiddleWords( List( portal.notes::link.notesID) ; $PR - 1 ; 1 );
        $$drag.portal = "notes"
        ];
        $PR - 1
        )

        /*
        $NewStart = $start - Position ( window::xSys_ContactNotes ; "•" ; 1 ; $PR - 1 ) - 2 ;
        $NewLength = $length
        */ ]
    Set Field [ window::xPt_ItemIDForScript; $link ]
    Set Field [ window.contact::Call Last Connect; join::link.date ]
    Set Field [ window::xPt_ItemIDForScript; "" ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Perform Script [ “UpdateLinks [ links ] ” ]
Else If [ LeftWords( Get(ScriptParameter) ; 1) = "delete" and
    Get ( ActiveFieldName ) = "xSys_ContactNotes" ]
    Set Variable [ $pos; Value:Let( [
        $start = Get ( ActiveSelectionStart ) ;
        $length = Get ( ActiveSelectionSize ) ;
        $item = "Å" ;
        $list =   Substitute ( Filter( Replace( $notes ; $start;  $length ;  $item) ; "•Å" ) ; "•" ; "¶" ) ;
        $PR = Position (  Filter( Substitute(  Substitute( "TTT" &  $list  & "TTT"; "¶" ; "TTT¶TTT" ) ; "TTT"  & $item  & "TTT" ; "#") ;
        "#¶"  ) ; "#" ;1; 1) ;
        $link = MiddleWords( List( portal.notes::link.notesID) ; $PR - 1 ; 1 );
        $$drag.portal = "notes"
        ];
        $PR - 1
        )

        /*
        $NewStart = $start - Position ( window::xSys_ContactNotes ; "•" ; 1 ; $PR - 1 ) - 2 ;
        $NewLength = $length
        */ ]
    Set Field [ window::xPt_ItemIDForScript; $link ]
    Show Custom Dialog [ Message: "Delete this note entry?"  & "¶¶" &

        Left(  Substitute (
        item.note::xSys_NoteFullField  ;
        [  "#dot#" & Char( 9 )  &  window.contact::xSys_LastNameFirstInitial   & "  " ; "#dot#" & Char( 9 )  ] ;
        [  "#dot#" ; "•" ] ;
        [  Char( 9 ) ; "  " ] ;
        [ "#start#" ; "" ]
        ) ; 330); Buttons: “OK”, “Cancel” ]
    If [ Get(LastMessageChoice) = 1 ]
        Perform Script [ “Move/Copy/ClearToAnyFolder [ $$drop.ID [1], $$DragList [2] cleans, g(sp) = from folders ] ”; Parameter: Let
            ( [ $$drop.id = $$trashID; $$draglist = $link] ; "") ]
    End If
    Set Field [ window::xPt_ItemIDForScript; "" ]
End If
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Exit Script [ Result: Let( [
    $$drop.note = "" ;
    $$drag.ID = "" ;
    $$drag.PortalID = "" ;
    $$drop.ID = "" ;
    $$drag.portal = "";
    $$drag.note = "";
    $$dragheadername = "";
    $$drop.sidebarID = "" ;
    $$drag.sidebarID = ""  ;
    $$Drag.foldertype = "" ;
    $$Draglist = "" ;
    $$DraglistPortalID = "" ;
    $$Dontchange = ""
    ] ; "" ) ]
Show Custom Dialog [ Message: Get ( ActiveSelectionStart )  & "¶" &
     Get ( ActiveSelectionSize ) & "¶" &  Length(window::xSys_ContactNotes ); Buttons: “OK”, “Cancel” ]
```

Window: old SwitchNotesViewTab [enter, exit, edit, connectdate] Copy

```
If [ LeftWords( Get(ScriptParameter) ; 1) = "enter" or LeftWords( Get(ScriptParameter) ; 1) = "exit" ]
    Set Field [ window::xSys_NotePictureEdit; Case(
        RightWords( Get(ScriptParameter) ;1 )  = "enter"; "✎";
        ""
        ) ]
    Set Field [ window::xSys_NotePictureConnectDate; Case(
        RightWords( Get(ScriptParameter) ;1 )  = "enter"; "☼";
        ""
        ) ]
    Set Selection [ window::xSys_ContactNotes; Start Position: Case(
        Length( window::xSys_ContactNotes ) < 2 ; 1;
        Get ( ActiveSelectionStart )  ≥ Length( window::xSys_ContactNotes ) ; 1 ;
        Get ( ActiveSelectionStart )
        ); End Position: Case(
        Get ( ActiveSelectionSize ) > 0 ; Get ( ActiveSelectionStart ) + Get ( ActiveSelectionSize ) - 1 ;
        ""
        ) ]
    Exit Script [  ]
End If
Set Variable [ $notes; Value:Substitute (
    List(  portal.notes::link.notes  ) ;
    [  "#dot#" & Char( 9 )  &  window.contact::xSys_LastNameFirstInitial   & "  " ; "#dot#" & Char( 9 )  ] ;
    [  "#dot#" & Char( 9 ) ;  Char( 9 ) ] ;
    [ "#start#" ; "•" ]
    ) ]
If [ GetLayoutObjectAttribute("tab.note.2";"isFrontTabPanel")  =  "1" ]
    If [ LeftWords( Get(ScriptParameter) ; 1) = "edit" and
        Get ( ActiveFieldName ) = "xSys_ContactNotes"

        //and Get ( ActiveSelectionSize ) > 0 ]
        Set Variable [ $pos; Value:Let( [
            $start = Get ( ActiveSelectionStart ) ;
            $length = Get ( ActiveSelectionSize ) ;
            $item = "Å" ;
            $list =   Substitute ( Filter( Replace( $notes ; $start;  $length ;  $item) ; "•Å" ) ; "•" ; "¶" ) ;
            $PR = Position (  Filter( Substitute(  Substitute( "TTT" &  $list  & "TTT"; "¶" ; "TTT¶TTT" ) ; "TTT"  & $item  & "TTT" ; "#") ;
            "#¶"  ) ; "#" ;1; 1) ;
            $link = MiddleWords( List( portal.notes::link.notesID) ; $PR - 1 ; 1 );
            $$drag.portal = "notes"
            ];
            $PR - 1
            )

            /*
            $NewStart = $start - Position ( window::xSys_ContactNotes ; "•" ; 1 ; $PR - 1 ) - 2 ;
            $NewLength = $length
            */ ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
        Perform Script [ “DoubleClicked [ link (optional), $$EnterNote , $$drag.portal]”; Parameter: $link ]
        Exit Script [  ]
        Go to Object [ Object Name: "field.note.note" ]
        Set Selection [ Start Position: If( $newStart < 1; 1; $newstart); End Position:  $newstart +

            $NewLength ]
    Else If [ LeftWords( Get(ScriptParameter) ; 1) = "connectdate" and
        Get ( ActiveFieldName ) = "xSys_ContactNotes"
        //and Get ( ActiveSelectionSize ) > 0 ]
        Set Variable [ $pos; Value:Let( [
            $start = Get ( ActiveSelectionStart ) ;
            $length = Get ( ActiveSelectionSize ) ;
            $item = "Å" ;
            $list =   Substitute ( Filter( Replace( $notes ; $start;  $length ;  $item) ; "•Å" ) ; "•" ; "¶" ) ;
            $PR = Position (  Filter( Substitute(  Substitute( "TTT" &  $list  & "TTT"; "¶" ; "TTT¶TTT" ) ; "TTT"  & $item  & "TTT" ; "#") ;
            "#¶"  ) ; "#" ;1; 1) ;
            $link = MiddleWords( List( portal.notes::link.notesID) ; $PR - 1 ; 1 );
            $$drag.portal = "notes"
            ];
            $PR - 1
            )
            /*
            $NewStart = $start - Position ( window::xSys_ContactNotes ; "•" ; 1 ; $PR - 1 ) - 2 ;
            $NewLength = $length
            */ ]
        Set Field [ window::xPt_ItemIDForScript; $link ]
        Set Field [ window.contact::Call Last Connect; join::link.date ]
        Set Field [ window::xPt_ItemIDForScript; "" ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
        Perform Script [ “UpdateLinks [ links ] ” ]
    Else If [ LeftWords( Get(ScriptParameter) ; 1) = "delete" and
        Get ( ActiveFieldName ) = "xSys_ContactNotes" ]
        Set Variable [ $pos; Value:Let( [
            $start = Get ( ActiveSelectionStart ) ;
            $length = Get ( ActiveSelectionSize ) ;
            $item = "Å" ;
            $list =   Substitute ( Filter( Replace( $notes ; $start;  $length ;  $item) ; "•Å" ) ; "•" ; "¶" ) ;
            $PR = Position (  Filter( Substitute(  Substitute( "TTT" &  $list  & "TTT"; "¶" ; "TTT¶TTT" ) ; "TTT"  & $item  & "TTT" ; "#") ;
            "#¶"  ) ; "#" ;1; 1) ;
            $link = MiddleWords( List( portal.notes::link.notesID) ; $PR - 1 ; 1 );
            $$drag.portal = "notes"
            ];
            $PR - 1
            )

            /*
            $NewStart = $start - Position ( window::xSys_ContactNotes ; "•" ; 1 ; $PR - 1 ) - 2 ;
            $NewLength = $length
            */ ]
        Perform Script [ “Move/Copy/ClearToAnyFolder [ $$drop.ID [1], $$DragList [2] cleans, g(sp) = from folders ] ”; Parameter: Let
            ( [ $$drop.id = $$trashID; $$draglist = $link] ; "") ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
    Else
        Go to Object [ Object Name: "tab.note.1" ]
    End If
Else
    Go to Object [ Object Name: "tab.note.2" ]
    Refresh Window
End If
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Exit Script [ Result: Let( [
    $$drop.note = "" ;
    $$drag.ID = "" ;
    $$drag.PortalID = "" ;
    $$drop.ID = "" ;
    $$drag.portal = "";
    $$drag.note = "";
    $$dragheadername = "";
    $$drop.sidebarID = "" ;
    $$drag.sidebarID = ""  ;
    $$Drag.foldertype = "" ;
    $$Draglist = "" ;
    $$DraglistPortalID = "" ;
    $$Dontchange = ""
    ] ; "" ) ]
Show Custom Dialog [ Message: Get ( ActiveSelectionStart )  & "¶" &
     Get ( ActiveSelectionSize ) & "¶" &  Length(window::xSys_ContactNotes ); Buttons: “OK”, “Cancel” ]
```

Window: SwitchContact [switch, double]

```
If [ LeftWords( Get(ScriptParameter) ; 1) = "switch" ]
    Set Variable [ $CID; Value:window.window::xPt_ContactID ]
    Set Variable [ $CL; Value:window.window::xPt_Opened_CompanyList ]
    Set Variable [ $CLL; Value:window.window::xPt_Opened_ContactLinkList ]
    Set Variable [ $N; Value:window.window::xPt_Opened_Notes ]
    Set Field [ window.window::xPt_ContactID; window::xPt_ContactID ]
    Set Field [ window.window::xPt_Opened_CompanyList; window::xPt_Opened_CompanyList ]
    Set Field [ window.window::xPt_Opened_ContactLinkList; window::xPt_Opened_ContactLinkList ]
    Set Field [ window.window::xPt_Opened_Notes; window::xPt_Opened_Notes ]
    Set Field [ window::xPt_ContactID; $CID ]
    Set Field [ window::xPt_Opened_CompanyList; $CL ]
    Set Field [ window::xPt_Opened_ContactLinkList; $CLL ]
    Set Field [ window::xPt_Opened_Notes; $N ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Perform Script [ “ItemTab” ]
End If
```

Window: MainList Sort [date field]

```
Freeze Window
Set Variable [ $sp; Value:Get(ScriptParameter) ]
If [ $sp = "date field" ]
    Set Variable [ $sp; Value:Case(
        window::xPt_SortMainListDateFieldChoice = "Connect" ; "Attempt" ;
        window::xPt_SortMainListDateFieldChoice = "Attempt"; "Date" ;
        "Connect"
        ) ]
    Set Field [ window::xPt_SortMainListDateFieldChoice; $sp ]
    Set Field [ window::xPt_SortMainListDateFieldNumber; Case(
        window::xPt_SortMainListDateFieldChoice = "Connect" ; "0" ;
        window::xPt_SortMainListDateFieldChoice = "Attempt"; "1" ;
        "2"
        ) ]
End If
Set Field [ window::xPt_SortMainListChoiceAscDesc; Case(
    window::xPt_SortMainlistChoice   ≠  $sp; window::xPt_SortMainListChoiceAscDesc ;
    window::xPt_SortMainListChoiceAscDesc = 1 ; "" ;
    1 )
    //asc = 1; desc = 0 ]
Set Field [ window::xPt_SortMainlistChoice; $sp ]
Set Field [ window::xPt_SortMainlistNumber; Case(
    window::xPt_SortMainlistChoice = "company" ; 0 ;
    window::xPt_SortMainlistChoice = "connect" and window::xPt_SortMainListChoiceAscDesc ; 1;
    window::xPt_SortMainlistChoice = "connect" ; 2;
    window::xPt_SortMainlistChoice = "attempt" and window::xPt_SortMainListChoiceAscDesc ; 3;
    window::xPt_SortMainlistChoice = "attempt" ; 4;
    window::xPt_SortMainlistChoice = "date" and window::xPt_SortMainListChoiceAscDesc ; 5;
    window::xPt_SortMainlistChoice = "date" ; 6;
    window::xPt_SortMainlistChoice = "reminder date"  and window::xPt_SortMainListChoiceAscDesc  ;  7;
    window::xPt_SortMainlistChoice = "reminder date" ;  8;
    window::xPt_SortMainlistChoice = "call category" ; 9  ;
    window::xPt_SortMainlistChoice = "reminder note" ; 10  ;
    window::xPt_SortMainlistChoice = "reminder flag" ; 11 ;
    window::xPt_SortMainlistChoice = "type" ;12    ;
    window::xPt_SortMainlistChoice = "owner" ;13 ;
    window::xPt_SortMainlistChoice = "description" ; 14  ;
    window::xPt_SortMainlistChoice = "contact" ;  15  ;
    0
    ) ]
If [ IsEmpty( window::xID_SpotlightEntry ) ]
    Set Field [ portal.sidebar.folder.selected::xFSelected_SortMainlistChoice; window::xPt_SortMainlistChoice ]
    Set Field [ portal.sidebar.folder.selected::xFSelected_SortMainListChoiceAscDesc; window::xPt_SortMainListChoiceAscDesc ]
End If
Perform Script [ “FolderSelect [ link, load, save ] ”; Parameter: window::xID_FolderSelectedID ]
Refresh Window
    [ Flush cached join results; Flush cached external data ]
Exit Script [  ]
Perform Script [ “UpdateLinks [ links ] ”; Parameter: window::xID_FolderSelectedID ]
```

Window: ItemTab

```
Go to Object [ Object Name: Case(
        portal.sidebar.folder.selected::Folder Type = "trash" ; "tab.topbar.trash";
        portal.sidebar.folder.selected::Folder Type = "inbox" ; "tab.topbar.inbox";
        portal.sidebar.folder.selected::Folder Type = "future" ; "tab.topbar.future";
        portal.sidebar.folder.selected::Folder Type = "removed" ; "tab.topbar.removed";
        portal.sidebar.folder.selected::Folder Type = "smart folder" ; "tab.topbar."  &  portal.sidebar.folder.selected::Folder SF type;
        portal.sidebar.folder.selected::Folder Type = "list" ; "tab.topbar."  &  portal.sidebar.folder.selected::Folder SF type;
        portal.sidebar.folder.selected::Folder Type = "folder" ; "tab.topbar.folder" ;
        "tab.topbar.blank"
        ) ]
Go to Object [ Object Name: Case(
        portal.sidebar.folder.selected::Folder Type = "trash" ; "tab.folder";
        portal.sidebar.folder.selected::Folder Type = "inbox" ; "tab.folder";
        portal.sidebar.folder.selected::Folder Type = "future" ; "tab.folder";
        portal.sidebar.folder.selected::Folder Type = "removed" ; "tab.folder";
        portal.sidebar.folder.selected::Folder Type = "folder" ; "tab.folder";
        "tab.folder.blank"
        ) ]
Set Variable [ $link; Value: Let( $l = Get(LayoutName) ;
        Case (
        $l = "Main Window" ; window::xPt_ContactID ;
        $l = "Contact Card" ; window::xPt_ContactID ;
        window::xPt_ItemID
        ) ) ]
Set Variable [ $lItemp; Value: window::xPt_ItemIDForScript ]
Set Field [ window::xPt_ItemIDForScript; $link ]
Set Window Title [ Current Window; New Title: "" ]
Set Window Title [ Current Window; New Title: Case (
        Get(LayoutName)  =  "Folder Options" ; Get(LayoutName);
        window::xPt_MainListLayout  =  "importemail" ; Get(LayoutName);
        Let( [
        $a = If ( PatternCount( Get(LayoutName) ;  "Main Window" ) or PatternCount( Get(LayoutName) ;  "Folder" ) ; portal.sidebar.folder.
        selected::Folder Name  & " - " ; "" ) &
        If ( IsEmpty ( join::link.type.window title ) ; Get ( LayoutName ) ; join::link.type.window title ) ;
        $b = PatternCount ( "¶" & WindowNames & "¶" ; "¶" & $a  & "¶")  ;
        $c = PatternCount ( "¶" & WindowNames ; "¶" & $a )
        ] ;
        Trim ( $a  &  If ( $b = 0  ; "" ;  " " & $c + 1 )  )
        )

        ) ]
Set Field [ window::xPt_ItemIDForScript; $lItemp ]
Exit Script [  ]
```

Window: SpotlightEntry [enter, close]

```
Freeze Window
Allow User Abort [ Off ]
If [   Get(ScriptParameter) = "enter" ]
    Perform Script [ “FolderSelect [ link, load, save ] ”; Parameter: window::xID_FolderSelectedID & " " & "save" ]
    Set Field [ window::xID_SpotlightEntry; "" ]
    Go to Object [ Object Name: "field.spotlight.entry" ]
Else If [   Get(ScriptParameter) = "close" ]
    If [ IsEmpty(window::xID_SpotlightEntry) ]
        If [ Get ( ActiveLayoutObjectName ) = "field.spotlight.entry" ]
            Commit Records/Requests
                [ Skip data entry validation; No dialog ]
            Perform Script [ “FolderSelect [ link, load, save ] ”; Parameter: window::xID_FolderSelectedID & " " & "load" ]
        Else
            Perform Script [ “FolderSelect [ link, load, save ] ”; Parameter: window::xID_FolderSelectedID & " " & "load" ]
            Go to Object [ Object Name: "field.spotlight.entry" ]
        End If
    Else
        If [ Get ( ActiveLayoutObjectName ) = "field.spotlight.entry" ]
            Perform Script [ “FolderSelect [ link, load, save ] ”; Parameter: window::xID_FolderSelectedID & " " & "load" ]
            Set Field [ window::xID_SpotlightEntry; "" ]
            Go to Object [ Object Name: "field.spotlight.entry" ]
        Else
            Set Field [ window::xID_SpotlightEntry; "" ]
            Commit Records/Requests
                [ Skip data entry validation; No dialog ]
            Perform Script [ “FolderSelect [ link, load, save ] ”; Parameter: window::xID_FolderSelectedID & " " & "load" ]
        End If
    End If
End If
Exit Script [ Result: Let( [
    $$QuickClick = "" ;
    $$drop.note = "" ;
    $$drag.ID = "" ;
    $$drag.PortalID = "" ;
    $$drop.ID = "" ;
    $$drag.portal = "";
    $$drag.note = "";
    $$dragheadername = "";
    $$drop.sidebarID = "" ;
    $$drag.sidebarID = ""  ;
    $$Drag.foldertype = "" ;
    $$Draglist = "" ;
    $$DraglistPortalID = "" ;
    $$Dontchange = ""
    ] ; "" ) ]
```

Window: SpotlightEntryModify

```
If [ Length( window::xID_SpotlightEntry ) > 2 ]
    Go to Object [ Object Name: "object.spotlight" ]
    Go to Object [ Object Name: "field.spotlight.entry" ]
End If
```

Window: GotoMainListItem [link & next, previous, same]

```
Set Variable [ $r; Value:Let( [
        $L =  List(portal.MainList::xxIDL_IDNumberPortal )  ;
        $lcount = If( IsEmpty( window::xID_SpotlightEntry ) ; portal.sidebar.folder.selected::Folder Count items ; WordCount( $L) ) ;
        $link = MiddleWords( Get(ScriptParameter) ; 1 ; 1) ;
        $sp = MiddleWords( Get(ScriptParameter) ; 2 ; 1) ;
        $PR = Position (  Filter( Substitute(  Substitute( "TTT" &  $L  & "TTT"; "¶" ; "TTT¶TTT" ) ; "TTT"  & $link  & "TTT" ; "#") ; "#¶"  ) ; "#" ;1;
        1)  ;

        $PRI = Case(
        IsEmpty( portal.sidebar.folder.selected::xFSelected_Selected_CurrentPortalRow ) ; 1;
        portal.sidebar.folder.selected::xFSelected_Selected_CurrentPortalRow < 1; 1;
        portal.sidebar.folder.selected::xFSelected_Selected_CurrentPortalRow > $lcount; $lcount ;
        portal.sidebar.folder.selected::xFSelected_Selected_CurrentPortalRow  ) ;

        $PR = If( IsEmpty( $PR) or $PR <  1 ; $PRI ; $PR) ;

        $PR = Case(
        $sp = "next" ; $PR + 1;
        $sp = "Previous" ; $PR - 1;
        $PR ) ;

        $PR = Case(
        $PR < 1 ; 1;
        $PR  ≥ $Lcount; $lcount
        $PR) ;

        $ac = MiddleWords( $L ; $PR ; 1 )
        ]; ""
        ) ]
Show Custom Dialog [ Message: $PR & "¶i" & $PRI; Buttons: “OK”, “Cancel” ]
Set Field [ window::xPt_SelectedItems_MainList; If( $sp = "next" or $sp = "previous" ; $ac ; window::xPt_SelectedItems_MainList ) ]
Set Field [ window::xPt_SelectedItems__CurrentClickedPortalRowMainlist; $PR ]
Go to Object [ Object Name: "portal.mainlist" ]
Go to Portal Row [ $PR -  2 ]
        [ Select; No dialog ]
Go to Portal Row [ $PR + 2 ]
        [ Select; No dialog ]
Go to Portal Row [ $PR ]
        [ Select; No dialog ]
```

Window: Folder: SmartFolderPref [pref, note]

```
Freeze Window
Set Variable [ $a; Value:Let( [
        $ac = window.contact::xxIDL_IDNumberInternal;
        $sp = MiddleWords(Get(ScriptParameter); 1; 1) ;
        $note = Trim(MiddleWords(Get(ScriptParameter); 2; 1)  & " "  & MiddleWords(Get(ScriptParameter); 3; 1) );
        $sk = Get ( ActiveModifierKeys )
        ];"")

        /*
        Shift = 1,
        Caps Lock = 2,
        Ctrl (Windows) and Control (Mac OS) = 4,
        Alt (Windows) and Option (Mac OS) = 8,
        ⌘ (Mac OS) = 16
        */ ]
If [ $sp = "action.type" ]
        Set Field [ portal.sidebar.folder.selected::xSF_ActionType;  Let( [ $a = "¶" & portal.sidebar.folder.selected::xSF_ActionType  & "¶" ; $t
        = "¶" & $note & "¶" ] ;
                Case(
                 PatternCount( $a  ; $t ) = 1 ; Substitute( $a ; [$t; "¶" ] ; ["¶¶" ; "¶" ]  ; ["¶¶" ; "¶" ] ) ;
                  $a  & $t
                ) ) ]
End If
If [ $sp = "action.completed" ]
        Set Field [ portal.sidebar.folder.selected::xSF_ActionCompleted; If( portal.sidebar.folder.selected::xSF_ActionCompleted = 1; 0; 1) ]
End If
Commit Records/Requests
        [ Skip data entry validation; No dialog ]
```

Window: Folder: EmptyTrash [all, selected]

```
Freeze Window
Set Field [ window::xPt_ItemIDForScript; "" ]
If [ IsEmpty( window::xID_SpotlightEntry ) ]
    If [ Get(ScriptParameter) = "selected" ]
        If [ WordCount(window::xPt_SelectedItems_MainList) > 0 ]
            Show Custom Dialog [ Title: "Delete"; Message: "Delete selected items?" & If(WordCount(window::
                xPt_SelectedItems_MainList) > 100 ; "¶¶(This operation may take several minutes)"; ""); Buttons: “Cancel”, “OK” ]
            If [ Get ( LastMessageChoice ) = 1 ]
                Exit Script [  ]
            End If
            Set Field [ window::xPt_Opened_MainList; "" ]
            Commit Records/Requests
                [ Skip data entry validation; No dialog ]
            Perform Script [ “Delete [links]”; Parameter: Filter( window::xPt_SelectedItems_MainList  ; List( portal.MainList::
                xxIDL_IDNumberPortal ) ) ]
        End If
    Else
        If [ WordCount(List( portal.MainList::xxIDL_IDNumberPortal )) > 0 ]
            Show Custom Dialog [ Title: "Delete"; Message: "Delete ALL items?" & If(WordCount( List( portal.MainList::
                xxIDL_IDNumberPortal ) ) > 300 ; "¶¶(This operation may take several minutes)"; ""); Buttons: “Cancel”, “OK” ]
            If [ Get ( LastMessageChoice ) = 1 ]
                Exit Script [  ]
            End If
            Set Field [ window::xPt_Opened_MainList; "" ]
            Commit Records/Requests
                [ Skip data entry validation; No dialog ]
            Perform Script [ “Delete [links]”; Parameter: List( portal.MainList::xxIDL_IDNumberPortal ) ]
        End If
    End If
End If
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Set Field [ portal.sidebar.folder.selected::xxIDL_LinkList; List( portal.MainList::xxIDL_IDNumberInternal ) ]
Perform Script [ “ItemTab”; Parameter: window::xPt_ContactID ]
Exit Script [ Result: Let( [
    $$drop.note = "" ;
    $$drag.ID = "" ;
    $$drag.PortalID = "" ;
    $$drop.ID = "" ;
    $$drag.portal = "";
    $$drag.note = "";
    $$dragheadername = "";
    $$drop.sidebarID = "" ;
    $$drag.sidebarID = ""  ;
    $$Drag.foldertype = "" ;
    $$Draglist = "" ;
    $$DraglistPortalID = "" ;
    $$Dontchange = ""
    ] ; "" ) ]
```

Window: Folder: FolderSelect [link, load, save]

```
Freeze Window
Set Field [ window::xPt_ItemIDForScript; "" ]
Set Field [ window::xID_SpotlightEntry; "" ]
Set Variable [ $a; Value:Let( [
    $sp =  Get(ScriptParameter);
    $link =  MiddleWords(  $sp; 1 ; 1 ) ;
    $note = MiddleWords( $sp; 2 ; 1 )
    ];"")

    /*
    Shift = 1,
    Caps Lock = 2,
    Ctrl (Windows) and Control (Mac OS) = 4,
    Alt (Windows) and Option (Mac OS) = 8,
    ⌘ (Mac OS) = 16
    */ ]
If [ $$lastedit = $link and Get(CurrentTime)  ≤ $$FolderClickTimer + 1 ]
    If [ portal.sidebar.folder::Folder Type = "Folder"
        or
        portal.sidebar.folder::Folder Type = "smart Folder" ]
        Go to Object [ Object Name: "field.folder" ]
        Exit Script [ Result: Let( $$EnterNote = "enter" ; "") ]
    End If
Else
    If [ Get(LayoutName)  ≠ "Folder Options" and
        Get(LayoutName)  ≠ "Preferences" ]
        If [ IsEmpty($note) or $note  = "save" ]
            Set Field [ portal.sidebar.folder.selected::xFSelected_SortMainlistChoice; window::xPt_SortMainlistChoice ]
            Set Field [ portal.sidebar.folder.selected::xFSelected_SortMainListChoiceAscDesc; window::
                xPt_SortMainListChoiceAscDesc ]
            Set Field [ portal.sidebar.folder.selected::xFSelected_Selected_ItemsMainList; window::xPt_SelectedItems_MainList ]
            Set Field [ portal.sidebar.folder.selected::xFSelected_Selected_CurrentPortalRow; window::
                xPt_SelectedItems__CurrentClickedPortalRowMainlist ]
            Set Field [ portal.sidebar.folder.selected::xFSelected_SortMainListDateFieldChoice; window::
                xPt_SortMainListDateFieldChoice ]
        End If
        #
        If [ IsEmpty($note) or $note  = "load" ]
            Perform Script [ “UpdateLinks [ links ] ”; Parameter: $link ]
            Set Field [ window::xID_FolderSelectedID; $link ]
            Set Field [ window::xID_FolderSlectedCategoryS; portal.sidebar.folder.selected::Folder Header CategoryS calc ]
            Set Field [ window::xPt_SortMainlistChoice; portal.sidebar.folder.selected::xFSelected_SortMainlistChoice ]
            Set Field [ window::xPt_SortMainListChoiceAscDesc; portal.sidebar.folder.selected::
                xFSelected_SortMainListChoiceAscDesc ]
            Set Field [ window::xPt_SelectedItems_MainList; Let( [
                $O = portal.sidebar.folder.selected::xFSelected_Selected_ItemsMainList ;
                $L =List( portal.MainList::xxIDL_IDNumberPortal ) ;
                $Lcount = WordCount ($L) ;
                $FO = FilterValues( $L ; $O)  ;
                $PRI = Case(
                IsEmpty( portal.sidebar.folder.selected::xFSelected_Selected_CurrentPortalRow ) ; 1;
                portal.sidebar.folder.selected::xFSelected_Selected_CurrentPortalRow < 1; 1;
                portal.sidebar.folder.selected::xFSelected_Selected_CurrentPortalRow > $lcount; $lcount ;
                portal.sidebar.folder.selected::xFSelected_Selected_CurrentPortalRow  )
                ] ;
                Case(
                WordCount( $FO ) > 0 ; $FO ;
                MiddleWords( $L ; $PRI ; 1 )
                )
                ) ]
            If [ WordCount( portal.sidebar.folder.selected::xFSelected_Selected_ItemsMainList )  > 0 ]
                Set Field [ window::xPt_SelectedItems__LastClickedPortal; "" ]
                Set Field [ window::xPt_SelectedItems__LastClickedPortalID; "" ]
                Set Field [ window::xPt_SelectedItems__CurrentClickedPortal; "mainlist" ]
                Set Field [ window::xPt_SelectedItems__CurrentClickedPortalID; LeftWords( window::xPt_SelectedItems_MainList ;
                        1) ]
                    Set Variable [ $r; Value:Let( [
                        $portalID =  window::xPt_SelectedItems__CurrentClickedPortalID   ;
                        $wc = WordCount( $L)  ;

                        $PRI = Position (  Filter( Substitute(  Substitute( "TTT" &  $L  & "TTT"; "¶" ; "TTT¶TTT" ) ; "TTT"  & $PortalID  &
                        "TTT" ; "#") ; "#¶"  ) ; "#" ;1; 1)

                        ];

                        ""
                        ) ]
                    Go to Object [ Object Name: "portal.mainlist" ]
                    Go to Portal Row [ Max( 1 ; $PRI - 4) ]
                        [ Select; No dialog ]
                    Go to Portal Row [ Case(
                        $PRI < 8 ; 1;
                        Min( $wc; $PRI + 4)
                        ) ]
                        [ Select; No dialog ]
                    Set Field [ window::xPt_SelectedItems__CurrentClickedPortalRowMainlist; $PRI ]
                End If
            End If
        Else
            Set Field [ window::xID_FolderSelectedID; $link ]
            Set Field [ window::xID_FolderSlectedCategoryS; portal.sidebar.folder.selected::Folder Header CategoryS calc ]
        End If
        Perform Script [ “ItemTab” ]
        Set Variable [ $$LastEdit; Value:$link ]
End If
```

Window: Folder: Move/Copy/ClearToAnyFolder [$$drop.ID [1], $$DragList [2] cleans, g(sp) = from…

```
#Dropbox [db=(drag, dragportal, drop, droptable, drop2 drop2portal/table) sp=(, drag2, drag3)]
Freeze Window
#
Set Variable [ $a; Value:Let( [
    $$drop.id =
    Case(
    IsEmpty( window::xSys_NewItem[1] ) ; $$drop.id ;
    window::xSys_NewItem[1] = "Inbox"; $$InboxID ;
    window::xSys_NewItem[1] = "Trash"; $$TrashID ;
    window::xSys_NewItem[1] = "Future"; $$FutureID ;
    If( Left( window::xSys_NewItem[1] ; 1) = "F" ;  window::xSys_NewItem[1] ; "" )
     ) ;

    $$DragList = If( IsEmpty( window::xSys_NewItem[1] ) ; $$DragList; window::xSys_NewItem[2] )

    ] ; ""  ) ]
If [ WordCount ($$draglist) > 0 ]
    #
    Set Variable [ $lltemp; Value:window::xPt_ItemIDForScript ]
    Set Field [ window::xPt_ItemIDForScript; $$draglist ]
    Set Variable [ $$draglist; Value:List( join::xxIDL_IDNumberInternal ) ]
    #
    #Remove from folder
    Set Field [ window::xPt_ItemIDForScript; $$trashID ]
    Set Variable [ $TL; Value:FilterValues( item.folder::xxIDL_LinkList ; $$draglist ) ]
    Set Variable [ $end; Value:Let(   [
        $LL =
        Case (
        WordCount( Get(ScriptParameter))  > 0 ; $$draglist ;
        $$drop.id = $$trashID ; $$DragList  ;
        $TL  )
        ;
        $count = 0
        ];
        WordCount($LL)  ) ]
    Loop
        Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
        Set Variable [ $link; Value: MiddleWords($LL; $count ; 1 ) ]
        Set Field [ window::xPt_ItemIDForScript; $link ]
        Set Field By Name [ join::xxIDL_FNTrash; Case(
            IsEmpty( $$drop.id ) ; 0 ;
            $$drop.id = $$trashID; 1;
            0 ) ]
        Set Field [ join::xxIDL_Trash; Case(
            IsEmpty( $$drop.id ) ; 0 ;
            $$drop.id = $$trashID; 1;
            0 ) ]
        Set Field [ item.reminder::xxIDL_Trash; Case(
            IsEmpty( $$drop.id ) ; 0 ;
            $$drop.id = $$trashID; 1;
            0 ) ]
        Set Variable [ $a; Value:Let(  [
            $count2 = 0  ;
            $LL2 =  If( $$drop.id = $$trashID; join::link.folderIDlist ; If( WordCount($TL ) = 0; ""; $$trashID )  & "¶" & Get
            (ScriptParameter) )  ;
            $end2 = WordCount( $LL2 )
            ] ; ""
            ) ]
        Loop
            Exit Loop If [ Let( $count2 = $count2 + 1 ; $count2)   >  $end2 or $end2 < 1 ]
            Set Field [ window::xPt_ItemIDForScript; MiddleWords($LL2;$count2;1) ]
            Set Field [ item.folder::xxIDL_LinkList; Substitute (item.folder::xxIDL_LinkList  ; [$link ; "" ] ; ["¶¶"; "¶"]) ]
        End Loop
        Set Field [ window::xPt_ItemIDForScript; $link ]
        If [ 0
            //WordCount( join::link.folderIDlist ) > 0 ]
            Set Field [ item.reminder::Date; "" ]
            Set Field [ item.reminder::Note; "" ]
            Set Field [ item.reminder::Flag; "" ]
            Set Field [ item.reminder::Time; "" ]
        End If
    End Loop
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    #
    #Add to new folder
    If [ Left($$drop.ID;1 )  = "f" ]
        Set Field [ window::xPt_ItemIDForScript; $$drop.id ]
        Set Field [ item.folder::xxIDL_LinkList; item.folder::xxIDL_LinkList & "¶" & $$draglist ]
        Set Field [ window::xPt_ItemIDForScript;  item.folder::xxIDL_LinkList ]
        Set Variable [ $cl; Value:List( join::xxIDL_IDNumberInternal ) ]
        Set Field [ window::xPt_ItemIDForScript; $$drop.id ]
        Set Field [ item.folder::xxIDL_LinkList; $cl ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
    End If
    #
    #Change reminder per folder settings
    Set Field [ window::xPt_ItemIDForScript; $$drop.id ]
    If [ Case(
        IsEmpty($$drop.id) ; "" ;
         not  IsEmpty( item.folder::xFolder_FolderOnDropPlusDays ) and
        item.folder::xFolder_FolderOnDropPlusDays  ≠ "Don't Change" ; 1 ;
         not  IsEmpty( item.folder::xFolder_FolderOnDropSetFlag ) and
        item.folder::xFolder_FolderOnDropSetFlag   ≠ "Don't Change" ; 1 ;
         not  IsEmpty( item.folder::xFolder_FolderOnDropChangeNote ) and
        item.folder::xFolder_FolderOnDropChangeNote   ≠ "Don't Change" ; 1 ;
        ""
        ) ]
        Set Variable [ $end; Value:Let(   [
            $LL = $$DragList  ;
            $count = 0
            ];
            WordCount($LL)  ) ]
        Loop
            Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
            Set Variable [ $link; Value: MiddleWords($LL; $count ; 1 ) ]
            Set Field [ window::xPt_ItemIDForScript; $$drop.id ]
            Set Field [ window::xID_ReminderID; $link ]
            Set Field [ window.reminder::Date; Let( nd =
                Case(
                IsEmpty( item.folder::xFolder_FolderOnDropPlusDays) or item.folder::xFolder_FolderOnDropPlusDays = "Don't
                Change" ;  window.reminder::Date ;
                item.folder::xFolder_FolderOnDropPlusDays = "Today"; Get(CurrentDate) ;
                item.folder::xFolder_FolderOnDropPlusDays = "Clear"; "" ;
                item.folder::xFolder_FolderOnDropPlusDays = "Exactly";  Get(CurrentDate) + GetAsNumber ( item.folder::
                xFolder_FolderOnDropPlusDaysTo ) ;

                item.folder::xFolder_FolderOnDropPlusDays = "If Empty Exactly";
                If( IsEmpty( window.reminder::Date ) ; Get(CurrentDate) + GetAsNumber ( item.folder::
                xFolder_FolderOnDropPlusDaysTo ) ; window.reminder::Date)  ;

                item.folder::xFolder_FolderOnDropPlusDays = "At least";
                Let( [ fd =  Get(CurrentDate) + GetAsNumber ( item.folder::xFolder_FolderOnDropPlusDaysTo ) ; rd = window.
                reminder::Date  ] ;

                Case (
                 fd > rd ; fd ;
                 rd
                )
                ) ;
                 window.reminder::Date
                )
                ;
                Case (
                IsEmpty( nd ); "" ;
                nd  = window.reminder::Date ; window.reminder::Date;
                DayName ( nd )  = "Saturday" ; nd + 2;
                DayName ( nd )  = "Sunday" ; nd + 1;
                nd
                )
                ) ]
            Set Field [ window.reminder::Time; Case(
                item.folder::xFolder_FolderOnDropPlusDays = "Today"; Get(CurrentTime) ;
                item.folder::xFolder_FolderOnDropPlusDays = "Clear"; "" ;
                item.folder::xFolder_FolderOnDropPlusDays = "Exactly"; Get(CurrentTime) ;
                item.folder::xFolder_FolderOnDropPlusDays = "Not Empty";   Get(CurrentTime) ;
                item.folder::xFolder_FolderOnDropPlusDays = "At least"; Get(CurrentTime) ;
                If( IsEmpty ( window.reminder::Time ) ; Get(CurrentTime)  ;  window.reminder::Time )
                ) ]
            Set Field [ window.reminder::Flag; Case(
                item.folder::xFolder_FolderOnDropSetFlag = "Don't change" or IsEmpty ( item.folder::
                xFolder_FolderOnDropSetFlag ) ; window.reminder::Flag ;
                item.folder::xFolder_FolderOnDropSetFlag = "Clear Blue"; If( window.reminder::Flag = "2"  or window.reminder::Flag =
                "0" ;  window.reminder::Flag  ; "") ;
                item.folder::xFolder_FolderOnDropSetFlag = "Clear all"; "" ;
                item.folder::xFolder_FolderOnDropSetFlag= "Blue"; If( window.reminder::Flag = "2"  or window.reminder::Flag = "0" ;
                window.reminder::Flag  ; "1") ;
                item.folder::xFolder_FolderOnDropSetFlag = "Red"; "2" ;
                item.folder::xFolder_FolderOnDropSetFlag = "Completed"; "0" ;
                window.reminder::Flag
                ) ]
            Set Field [ window.reminder::Note; If( IsEmpty( item.folder::xFolder_FolderOnDropChangeNote ) or item.folder::
            xFolder_FolderOnDropChangeNote = "Don't Change" ; window.reminder::Note;

                Let( nn =

                Trim ( item.folder::xFolder_FolderOnDropChangeNoteTo  & " "  &
                Substitute  (
                item.folder::xFolder_FolderOnDropChangeNoteAdder1 & " "  &
                item.folder::xFolder_FolderOnDropChangeNoteAdder2 & " "  &
                item.folder::xFolder_FolderOnDropChangeNoteAdder3
                ; [ "Folder Name" ;  item.folder::Folder Name ]
                ; [ "Day Name" ; DayName ( window.item.reminder::Date ) ]
                ; [ "Month Name" ; MonthName ( window.item.reminder::Date ) ]
                ; [ "Year" ; Left( Year ( window.item.reminder::Date ); 2)  ]
                ; [ "None" ; "" ]
                )
                )
                ;
                Case (
                item.folder::xFolder_FolderOnDropChangeNote = "If Empty" ;
                If ( IsEmpty( window.reminder::Note ) ; nn ; window.reminder::Note ) ;

                item.folder::xFolder_FolderOnDropChangeNote = "Always" ;
                nn ;

                item.folder::xFolder_FolderOnDropChangeNote = "If Not Empty Add" ;
                If ( IsEmpty( window.reminder::Note ) ; nn ; Trim (window.reminder::Note  & " " &   nn )   ) ;

                item.folder::xFolder_FolderOnDropChangeNote = "Always Add" ;
                Trim (window.reminder::Note  & " " &   nn ) ;

                item.folder::xFolder_FolderOnDropChangeNote = "Clear" ;
                "" ;
                window.reminder::Note
                )
                )) ]
            Commit Records/Requests
                [ Skip data entry validation; No dialog ]
            Set Field [ window::xPt_ItemIDForScript; $link ]
            Set Field [ join::link.sort.field.modifedtimestamp; Get(CurrentTimeStamp) ]
            Commit Records/Requests
                [ Skip data entry validation; No dialog ]
        End Loop
    End If
    #
```

```
    #
End If
#
Set Field [ window::xPt_ItemIDForScript; $lItemp ]
#
If [ IsEmpty( window::xSys_NewItem ) ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Exit Script [  ]
Else
    Set Field [ window::xSys_NewItem; "" ]
    Set Field [ window::xSys_NewItem[2]; "" ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Exit Script [ Result: Let( [
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drag.PortalID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$DraglistPortalID = "" ;
        $$Dontchange = ""
        ] ; "" ) ]
End If
Show Custom Dialog [ Message: "drag:"  & $$drag.ID & "¶drop " &  $$drop.ID & "¶fail " & $$fail & "¶note: " & $$drop.note & "¶drop.siID: "
& $$drop.sidebarID
    & "¶drag.siID: " & $$drag.sidebarID
    & "¶portal: " & $$drag.portal
    & "¶drag.portalID: " & $$drag.portalID
    & "¶dragnote: " & $$drag.note
    & "¶draglist: " & $$draglist
    & "¶dontchange: " & $$dontchange
    & "¶link: " & $$link
    & "¶item: " & $$item
    & "¶editlink: " & window::xPt_ItemID
    & "¶editlinkCalc: " & window::xPt_ItemIDcalc
    & "¶editlinkForScript: " & window::xPt_ItemIDForScript
    & "¶clikdedIDs: " & window::xPt_SelectedItems__CurrentClickedPortalSelectedIDs
    & "¶clickedportal: " & window::xPt_SelectedItems__CurrentClickedPortal
    & "¶clickedLINK: " & window::xPt_SelectedItems__CurrentClickedPortalID
    & "¶clickedFN: " & window::xPt_SelectedItems__CurrentClickedPortalFN
    & "¶clikdedIDs: " & window::xPt_SelectedItems__CurrentClickedPortalSelectedIDs; Buttons: “OK”, “Cancel” ]
Show Custom Dialog [ Title: 3; Message: "draglist: " & $$draglist; Buttons: “OK”, “Cancel” ]
```

Window: Drag, Drop, and Selctiong: Dropbox

```
Freeze Window
#
#--- Click ---
If [ IsEmpty( $$drop.note ) ]
    If [ $$QuickClick and
        Get ( ActiveModifierKeys )  = 0 and
        window::xPt_SelectedItems__CurrentClickedPortalID  ≠  $$drag.PortalID and
        IsEmpty(window::xID_SpotlightEntry) and
        $$drag.portal  = "mainlist" and
        window::xPt_MainListLayout  ≠  "importemail" ]
        Perform Script [ “QuickClicked ” ]
    Else
        Perform Script [ “Clicked [$$drag.PortalID $$drag.portal, "opensub"]” ]
    End If
    Exit Script [ Result: Let( [
        $$QuickClick = "" ;
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drag.PortalID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$DraglistPortalID = "" ;
        $$Dontchange = ""
        ] ; "" ) ]
End If
#
#--- Drop on list ---
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Set Variable [ $$drop.id; Value:Case(
    Left( $$drag.id;1)   =  "F" ; $$drop.id ;

    $$drop.note = "No MainList" and
    $$drag.portal   ≠  "MainList" and
    portal.sidebar.folder.selected::Folder Type  ≠  "Smart Folder" and
    portal.sidebar.folder.selected::Folder Type  ≠  "List" and
    IsEmpty(window::xID_SpotlightEntry) and
    not IsEmpty( window::xID_FolderSelectedID) ; window::xID_FolderSelectedID ;

    $$drop.note = "No ContactLinkList" and
    $$drag.portal   ≠  "ContactLinkList" and
    not IsEmpty(window::xPt_ContactID) ; window::xPt_ContactID ;

    $$drop.note = "No ContactLinkList2" and
    $$drag.portal   ≠  "ContactLinkList2" and
    not IsEmpty(window::xPt_ContactID) ; window::xPt_ContactID ;

    $$drop.note = "No Notes" and
    $$drag.portal   ≠  "Notes" and
    not IsEmpty(window::xPt_ContactID) ; window::xPt_ContactID ;

    $$drop.note = "No ItemLinkList" and
    Left( $$drag.id;1)   ≠  "F" and
    $$drag.portal   ≠  "ItemLinkList" and
    not IsEmpty(  window::xPt_ItemID  ); window::xPt_ItemID ;

    $$drop.note = "No CompanyList" and
    $$drag.portal   ≠  "CompanyList" and
    not IsEmpty(window::xPt_ContactID) ; window::xPt_ContactID ;
    $$drop.id
    ) ]
Set Variable [ $$drop.note; Value:Case(
    Left( $$drag.id;1)   =  "F" ; $$drop.note;

    $$drop.note = "No MainList" and
    $$drag.portal   ≠  "MainList" and
    portal.sidebar.folder.selected::Folder Type  ≠  "Smart Folder" and
    portal.sidebar.folder.selected::Folder Type  ≠  "List" and
    IsEmpty(window::xID_SpotlightEntry) and
    not IsEmpty( window::xID_FolderSelectedID) ; portal.sidebar.folder.selected::Folder Type ;

    $$drop.note = "No ContactLinkList" and
    $$drag.portal   ≠  "ContactLinkList" and
    not IsEmpty(window::xPt_ContactID) ; ""  ;

    $$drop.note = "No ContactLinkList2" and
    $$drag.portal   ≠  "ContactLinkLis2t" and
    not IsEmpty(window::xPt_ContactID) ; ""  ;

    $$drop.note = "No Notes" and
    $$drag.portal   ≠  "Notes" and
    not IsEmpty(window::xPt_ContactID) ; ""  ;

    $$drop.note = "No ItemLinkList" and
    Left( $$drag.id;1)   ≠  "F" and
    $$drag.portal   ≠  "ItemLinkList" and
    not IsEmpty(  window::xPt_ItemID  ); "itemlinklist" ;

    $$drop.note = "No CompanyList" and
    $$drag.portal   ≠  "CompanyList" and
    not IsEmpty(window::xPt_ContactID) ; "" ;

    $$drop.note
    ) ]
#
Set Field [ window::xPt_SelectedItems__CurrentClickedPortalSelectedIDs; Let( a = FilterValues (
    Evaluate ( "xPt_SelectedItems_" & window::xPt_SelectedItems__CurrentClickedPortal)  & "¶" ;
    List( Evaluate( "portal." & window::xPt_SelectedItems__CurrentClickedPortal & "::xxIDL_IDNumberPortal" )   )
     )

    ;
    Substitute( Trim( Substitute ( a ; "¶" ; " "  ) ) ; " " ; "¶" )
    ) ]
Set Variable [ $sk; Value:Get ( ActiveModifierKeys )


    /*
    Shift = 1,
    Caps Lock = 2,
    Ctrl (Windows) and Control (Mac OS) = 4,
    Alt (Windows) and Option (Mac OS) = 8,
    ⌘ (Mac OS) = 16
    */ ]
Set Variable [ $$DragList; Value:If(
    PatternCount ( window::xPt_SelectedItems__CurrentClickedPortalSelectedIDs; $$drag.PortalID ) > 0 ;
    window::xPt_SelectedItems__CurrentClickedPortalSelectedIDs ;
    $$drag.PortalID
    ) ]
#
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
If [ PatternCount($$drop.note; "no " ) > 0 or
    $$drag.note = "header" or
    $$drop.note = "header" or

    $$drag.note = "Desktop" or
    $$drag.note = "Users" ]
Else If [ LeftWords( $$drag.ID ; 1)   =  LeftWords( $$drop.ID  ; 1) ]
Else If [ $$drop.note= "folder" and Left( $$drag.id;1)  ≠ "F" ]
    #--- To Folder ---
    Perform Script [ “Move/Copy/ClearToAnyFolder [ $$drop.ID [1], $$DragList [2] cleans, g(sp) = from folders ] ”; Parameter: If( $$drag.
        portal = "mainlist" ; If ($sk = 1 or $sk = 4 or $sk = 16 or $$futureID = $$drop.id ; window::xID_FolderSelectedID; "" ) ; "" ) ]
Else If [ $$drag.note & $$drop.note = "Smart FoldersFolder" or
    $$drag.note & $$drop.note = "FoldersFolder" or
    $$drag.note & $$drop.note = "FoldersSmart Folder" or
    $$drag.note & $$drop.note = "ListsList" ]
    #--- folder to folder change position ---
    Set Field [ window::xPt_ItemIDForScript; $$drop.id ]
    Set Variable [ $drop.sidebar.ts; Value:item.folder::Folder Sort ]
    Set Variable [ $drop.parent; Value:item.folder::Folder folder parent ]
    Set Field [ window::xPt_ItemIDForScript; $$drag.PortalID ]
    Set Field [ item.folder::Folder folder parent; Case (
        $sk = 1  or $sk = 4 or $sk = 16 ;
        Case(
        not IsEmpty( item.folder::Folder arrow down); "" ;
        IsEmpty( $drop.parent ) ; $$drop.id;
        $drop.parent
        ) ;

        $drop.parent
        ) ]
    Set Field [ item.folder::Folder Sort; Case (
        $sk = 1  or $sk = 4 or $sk = 16 ;
        Case(
        not IsEmpty( item.folder::Folder arrow down); item.folder::Folder Sort  ;
        $drop.sidebar.ts  + .1
        );
        item.folder::Folder Sort > $drop.sidebar.ts ; $drop.sidebar.ts  - .1 ;
        $drop.sidebar.ts  + .1
        ) ]
    Set Field [ window::xID_FolderHeaderAllForScript; "All" ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Set Variable [ $a; Value:Let(  [
        $count = 0  ;
        $LL = List(portal.sidebar.folder::xxIDL_IDNumberInternal) ;
        $end = WordCount( $LL )
        ] ; ""
        ) ]
    Loop
        Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
        Set Field [ window::xPt_ItemIDForScript; MiddleWords($LL;$count;1) ]
        Set Field [ join::xxIDL_IDNumberPortal; MiddleWords($LL;$count;1) ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
        Set Field [ item.folder::Folder Sort; If ( IsEmpty( item.folder::Header Name ) ; $count ; 0) ]
    End Loop
    Set Field [ window::xID_FolderHeaderAllForScript; "" ]
Else If [ Case(
    $$drop.note = "trash"  and $$drag.note = "folders" ; 1;
    $$drop.note = "trash"  and $$drag.note = "smart folders" ; 1;
    ""
    ) ]
    Show Custom Dialog [ Message: "Delete Folder?"; Buttons: “Cancel”, “OK” ]
    Perform Script [ “Delete [links]”; Parameter: If ( Get ( LastMessageChoice ) = "2"; $$drag.PortalID ; "" ) ]
Else If [ $$drop.note = "trash"  and $$drag.portal = "portal.sidebar" ]
    Perform Script [ “Delete [links]”; Parameter: $$drag.PortalID ]
Else If [ Case(
    Left($$drag.id ; 1)  =  "F" ; "" ;
    $$drop.id = $$futureID ; 1 ;
    "" ) ]
    #--- Drop on future ---
    Set Variable [ $$cfd; Value:"" ]
    New Window [ Name: "Future Date"; Height: 97; Width: 454; Top: 167; Left: 436 ]
    Show/Hide Status Area
        [ Hide ]
    Move/Resize Window [ Current Window; Height: 97; Width: 454; Top: 167; Left: 436 ]
    Go to Layout [ “Future Date” (window) ]
    Freeze Window
    Set Field [ window::xID_ReminderID;  MiddleWords($$DragList ; 1 ; 1 ) ]
    Set Field [ window::xSys_TempFutureDate; "" ]
    Set Field [ window::xSys_TempFutureNote; window.reminder::Note ]
    Set Variable [ $tempdate; Value:"" ]
    Set Variable [ $tempnote; Value:"" ]
    Set Field [ window::xID_ReminderID; "" ]
    Go to Field [ window::xSys_TempFutureDate ]
        [ Select/perform ]
    If [ 0 ]
        Allow User Abort [ Off ]
    End If
    Pause/Resume Script [ Indefinitely ]
    Allow User Abort [ On ]
    Set Variable [ $tempdate; Value:window::xSys_TempFutureDate ]
    Set Variable [ $tempnote; Value:window::xSys_TempFutureNote ]
    Close Window [ Current Window ]
    If [ $$cfd  ≠  1 ]
        Set Variable [ $tempdate; Value:If(IsEmpty( $tempdate ) ; Get(CurrentDate) + 1; $tempdate) ]
        Set Variable [ $end; Value:Let(   [
            $LL = $$DragList  ;
            $count = 0
            ];
            WordCount($LL)  ) ]
        Loop
            Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
            Set Variable [ $link; Value: MiddleWords($LL; $count ; 1 ) ]
            Set Field [ window::xID_ReminderID; $link ]
            Set Field [ window.reminder::Date; Case(
                IsEmpty( $tempdate ) and IsEmpty( window.reminder::Date ) ; Get(CurrentDate) + 1;
                IsEmpty( $tempdate ) and window.reminder::Date  < Get(CurrentDate) ; Get(CurrentDate) + 1;
                IsEmpty( $tempdate ) ; window.reminder::Date  ;
                $tempdate
                ) ]
            Set Field [ window.reminder::Note; Case(
                IsEmpty( $tempnote ) ;  window.reminder::Note ;
                $tempnote
                ) ]
        End Loop
        Set Field [ window::xID_ReminderID; "" ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
        Perform Script [ “Move/Copy/ClearToAnyFolder [ $$drop.ID [1], $$DragList [2] cleans, g(sp) = from folders ] ”; Parameter: Let(
            $$Drop.Id = $$drop.id ;
            Case (
            $$drop.id = $$futureID and $$drag.portal = "mainlist" ;  window::xID_FolderSelectedID ;
            $sk = 1 and $$drag.portal = "mainlist" ;  window::xID_FolderSelectedID ;
            $sk = 4 and $$drag.portal = "mainlist" ;  window::xID_FolderSelectedID ;
            $sk = 16 and $$drag.portal = "mainlist" ;  window::xID_FolderSelectedID ;
             "") ) ]
    End If
Else If [ Case(
    Left($$drag.id ; 1)  =  "F" ; "" ;
    $$drop.note= "inbox"  ; 1 ;
    $$drop.note= "Trash" ; 1 ;
    "" ) ]
    Perform Script [ “Move/Copy/ClearToAnyFolder [ $$drop.ID [1], $$DragList [2] cleans, g(sp) = from folders ] ”; Parameter: Let(
        $$Drop.Id = $$drop.id ;
        Case (
        $$drop.id = $$futureID and $$drag.portal = "mainlist" ;  window::xID_FolderSelectedID ;
        $sk = 1 and $$drag.portal = "mainlist" ;  window::xID_FolderSelectedID ;
        $sk = 4 and $$drag.portal = "mainlist" ;  window::xID_FolderSelectedID ;
        $sk = 16 and $$drag.portal = "mainlist" ;  window::xID_FolderSelectedID ;
         "") ) ]
Else If [ $$drop.note = "sidebarnew" and
    Case(
    $$drag.portal = "portal.sidebar" ; "" ;
    Left($$drag.ID ;1 )   =   "F" ; "" ;
    Left($$drag.PortalID ;1 )   =   "F" ; "" ;
    PatternCount( List( portal.sidebar::xSidebar_IDitemNumberInternal ); $$drag.id ) > 0 ; "";
    1) ]
    #--- New sidebar item ---
    #--- New sidebar item ---
    New Window [ Height: 1; Width: 1; Top: -400 ]
    Go to Layout [ “data.folder” (item.folder) ]
    New Record/Request
    Set Variable [ $link; Value:item.folder::xxIDL_IDNumberInternal ]
    Set Field [ item.folder::xSidebar_IDitemNumberInternal; $$drag.id ]
    Set Field [ item.folder::xID_Sidebar; 1 ]
    Close Window [ Current Window ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Perform Script [ “LinkItems [$$drop.id[1], $$draglist[2] ] / NewLinks [ $$drop.id [1] ]”; Parameter: Let( [ $$drop.id = $link ; $$draglist =
        "" ] ; "") ]
Else If [ $$drop.note = "sidebar" and $$drag.portal = "portal.sidebar" ]
    #--- sidebar to sidebar change position ---
    Set Field [ window::xPt_ItemIDForScript; $$drop.sidebarID ]
    Set Variable [ $drop.sidebar.ts; Value:item.folder::xSidebar_Sort ]
    Set Field [ window::xPt_ItemIDForScript; $$drag.sidebarID ]
    Set Field [ item.folder::xSidebar_Sort; If( item.folder::xSidebar_Sort > $drop.sidebar.ts ; $drop.sidebar.ts  - .1 ; $drop.sidebar.ts
        + .1 ) ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Set Variable [ $a; Value:Let(  [
        $count = 0 ;
        $LL = List(portal.sidebar::xxIDL_IDNumberInternal) ;
        $end = WordCount( $LL )
        ] ; ""
        ) ]
    Loop
        Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
        Set Field [ window::xPt_ItemIDForScript; MiddleWords($LL;$count;1) ]
        Set Field [ item.folder::xSidebar_Sort; $count ]
    End Loop
Else If [ $$drop.note = "sidebar" ]
    #--- drop onto existing sidebar---
    If [ $$drag.portal = "portal.sidebar" or Left($$drag.PortalID ;1 )   =   "F" or Left($$drag.ID ;1 )   =   "F" or PatternCount( List( portal.
        sidebar::xSidebar_IDitemNumberInternal ); $$drag.id ) > 0 ]
    Else
        Set Field [ window::xPt_ItemIDForScript; $$drop.sidebarID ]
        Set Field [ item.folder::xSidebar_IDitemNumberInternal; $$drag.id ]
    End If
Else If [ not IsEmpty($$drop.ID )
    and Left($$drop.ID; 1)  ≠ "F"
    and Left($$drag.ID; 1)  ≠ "F"
    and Left($$drag.PortalID; 1)  ≠ "F" ]
    #--- Normal dropbox linking to items together ---
    Set Field [ window::xPt_ItemIDForScript;  MiddleWords($$Drag.id ; 1 ; 1 ) ]
    If [ WordCount ( FilterValues( join::xxIDL_LinkListTotal ; $$drop.id ) ) > 0 and
        WordCount ($$draglist ) = 1 ]
        Set Field [ window::xPt_ItemIDForScript; "" ]
        Set Variable [ $$draglist; Value:$$drag.id ]
        Perform Script [ “LinkRemove [$$drop.id, $$draglist, cleans]” ]
    Else
        Set Field [ window::xPt_ItemIDForScript; "" ]
        Perform Script [ “LinkItems [$$drop.id[1], $$draglist[2] ] / NewLinks [ $$drop.id [1] ]” ]
    End If
End If
Set Field [ window::xPt_ItemIDForScript; "" ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
```

```
Exit Script [ Result: Let( [
        $$QuickClick = "" ;
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drag.PortalID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$DraglistPortalID = "" ;
        $$Dontchange = ""
        ] ; "" ) ]
Show Custom Dialog [ Message: "drag:"  & $$drag.ID & "¶drop " &  $$drop.ID & "¶fail " & $$fail & "¶note: " & $$drop.note & "¶drop.siID: "
& $$drop.sidebarID
        & "¶drag.siID: " & $$drag.sidebarID
        & "¶portal: " & $$drag.portal
        & "¶drag.portalID: " & $$drag.portalID
        & "¶dragnote: " & $$drag.note
        & "¶draglist: " & $$draglist
        & "¶dontchange: " & $$dontchange
        & "¶link: " & $$link
        & "¶item: " & $$item
        & "¶editlink: " & window::xPt_ItemID
        & "¶editlinkCalc: " & window::xPt_ItemIDcalc
        & "¶editlinkForScript: " & window::xPt_ItemIDForScript
        & "¶clikdedIDs: " & window::xPt_SelectedItems__CurrentClickedPortalSelectedIDs
        & "¶clickedportal: " & window::xPt_SelectedItems__CurrentClickedPortal
        & "¶clickedLINK: " & window::xPt_SelectedItems__CurrentClickedPortalID
        & "¶clickedFN: " & window::xPt_SelectedItems__CurrentClickedPortalFN
        & "¶clikdedIDs: " & window::xPt_SelectedItems__CurrentClickedPortalSelectedIDs; Buttons: “OK”, “Cancel” ]
```

Window: Drag, Drop, and Selctiong: Clicked [$$drag.PortalID $$drag.portal, "opensub"]

```
Freeze Window
If [ Get(ScriptParameter) = "opensub" ]
    Set Field [ window::xID_FolderHeaderParents; Let( [  $L =  window::xID_FolderHeaderParents ; $FID = portal.sidebar.folder::
    xxIDL_IDNumberInternal ] ;

        Case(
        IsEmpty( portal.sidebar.folder::Folder arrow down) ; $L ;
        PatternCount( "¶" & $L & "¶" ; "¶"  & $FID & "¶") > 0 ; Substitute("¶" &  $L  & "¶"; ["¶" &  $FID & "¶" ; "¶" ] ; ["¶¶" ; "¶" ] ; ["¶¶" ;
        "¶" ] ) ;
        $L  & "¶" & $FID
        )
        ) ]
    Set Field [ window.user::xPrefUser_FolderHeaderParents; window::xID_FolderHeaderParents ]
    Halt Script
End If
Set Variable [ $drag.portal; Value:Case(
    IsEmpty(window::xID_SpotlightEntry) and window::xPt_SelectedItems__CurrentClickedPortal = "MainList"; "MainList" ;
    not IsEmpty(window::xID_SpotlightEntry) and window::xPt_SelectedItems__CurrentClickedPortal = "MainList"; "Spotlight" ;
    $$drag.portal
    ) ]
Set Field [ window::xPt_SelectedItems__LastClickedPortal; window::xPt_SelectedItems__CurrentClickedPortal ]
Set Field [ window::xPt_SelectedItems__LastClickedPortalID; window::xPt_SelectedItems__CurrentClickedPortalID ]
If [ Left($$drag.id ; 1)  ≠ "F" and $$drag.portal  ≠ "folder.sidebar" ]
    Set Field [ window::xPt_SelectedItems__CurrentClickedPortal; $$drag.portal ]
    Set Field [ window::xPt_SelectedItems__CurrentClickedPortalID; $$Drag.PortalID ]
    Set Field [ window::xPt_SelectedItems__CurrentClickedPortalRowMainlist; Case(
        window::xPt_SelectedItems__CurrentClickedPortal = "MainList"  ; Get ( ActivePortalRowNumber ) ;
        window::xPt_SelectedItems__CurrentClickedPortalRowMainlist
        ) ]
Else
    Set Field [ window::xPt_SelectedItems__CurrentClickedPortal; $$drag.portal ]
    Set Field [ window::xPt_SelectedItems__CurrentClickedPortalID; "" ]
End If
#
#--- Item Select ---
If [ $$drag.portal = "portal.sidebar.folder" ]
    If [ $$drag.note  =  "header" ]
        Set Field [ window::xID_FolderHeaderTypeExclude; Substitute (

            If( PatternCount ("¶" &  Upper( window::xID_FolderHeaderTypeExclude)  & "¶"; "¶" & $$dragheadername  & "¶" ) > 0 ;
            Substitute( "¶" &  Upper( window::xID_FolderHeaderTypeExclude) & "¶" ; "¶" & $$dragheadername  & "¶" ; "¶" ) ;
             "¶" & $$dragheadername  & "¶"  & window::xID_FolderHeaderTypeExclude  & "¶"
            )

            ; ["¶¶" ; "¶" ] ; ["¶¶" ; "¶" ] ; ["¶¶" ; "¶" ]  ) ]
        Set Field [ window::xID_FolderHeaderTypeExclude; If(WordCount( window::xID_FolderHeaderTypeExclude ) = 0  ; "1";
            window::xID_FolderHeaderTypeExclude ) ]
        Set Field [ window.user::xPrefUser_FolderHeaderTypeExclude; window::xID_FolderHeaderTypeExclude ]
        Set Field [ window.user::xPrefUser_FolderHeaderParents; window::xID_FolderHeaderParents ]
    Else
        If [ not IsEmpty( window::xID_SpotlightEntry ) ]
            Set Field [ window::xID_SpotlightEntry; "" ]
            Perform Script [ “FolderSelect [ link, load, save ] ”; Parameter: $$drag.id  &  " load" ]
        Else
            Perform Script [ “FolderSelect [ link, load, save ] ”; Parameter: $$drag.id ]
        End If
        Set Variable [ $$FolderClickTimer; Value:Get(CurrentTime) ]
    End If
Else If [ $$drag.portal = "portal.sidebar" and not IsEmpty($$drag.id) ]
    Perform Script [ “SidebarItemSelect [link ]”; Parameter: $$drag.id ]
    Set Variable [ $$FolderClickTimer; Value:Get(CurrentTime) - 2 ]
Else If [ window::xPt_MainListLayout = "importemail" and $$Drag.Portal = "LinkToItemList" ]
```

```
    Perform Script [ “LinkItems [$$drop.id[1], $$draglist[2] ] / NewLinks [ $$drop.id [1] ]”; Parameter: Let (  [$$drop.id = window::
        xPt_ItemID ; $$draglist = $$Drag.ID]; "" ) ]
    If [ Left($$drag.id;1) = "C" ]
        Set Field [ window::xPt_ContactID; $$drag.id ]
        Set Field [ window::xPt_Opened_ContactLinkList; "" ]
    End If
Else If [ Case(
    $$drag.portal  =  "Item" ; 0 ;
    $$drag.portal  =  "contact"  and Get(LayoutName)  = "Contact Card"; 0 ;
    1) ]
    Perform Script [ “Clicked Part2” ]
End If
If [ $$EnterNote ≠  "enter" ]
    Perform Script [ “ItemTab” ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
End If
Set Variable [ $$EnterNote; Value:"" ]
```

Window: Drag, Drop, and Selctiong: Clicked Part2

```
Set Field [ window::xPt_ItemIDForScript; "" ]
Set Variable [ $a; Value:Let( [
        $sp =  Get(ScriptParameter);
        $PortalID =  window::xPt_SelectedItems__CurrentClickedPortalID  ;
        $sk = Get ( ActiveModifierKeys )
        ];"")

        /*
        Shift = 1,
        Caps Lock = 2,
        Ctrl (Windows) and Control (Mac OS) = 4,
        Alt (Windows) and Option (Mac OS) = 8,
        ⌘ (Mac OS) = 16
        */ ]
#
If [ $sk = 16 or $sk = 1 or $sk = 8 ]
        If [ $sk = 16 ]
                #-- Option Click --
                Set Field By Name [ window::xPt_SelectedItems__CurrentClickedPortalFN; If(
                        PatternCount( window::xPt_SelectedItems__CurrentClickedPortalSelectedIDs ; window::
                        xPt_SelectedItems__CurrentClickedPortalID ) = 0 ; $PortalID  & "¶" ;
                        "" )  &
                        Substitute ( window::xPt_SelectedItems__CurrentClickedPortalSelectedIDs ; [ window::
                        xPt_SelectedItems__CurrentClickedPortalID ; "" ];["¶¶";"¶"]  ) ]
        Else If [ $sk = 1 ]
                #-- Shift Click --
                Set Field By Name [ window::xPt_SelectedItems__CurrentClickedPortalFN; Let( [
                        $portalID = window::xPt_SelectedItems__CurrentClickedPortalID;
                        $list = List( Evaluate( "portal." & window::xPt_SelectedItems__CurrentClickedPortal & "::xxIDL_IDNumberPortal" )  ) ;
                        $listclean = List( Evaluate( "portal." & window::xPt_SelectedItems__CurrentClickedPortal & "::
                        xxIDL_IDNumberInternal" )  ) ;
                        $sl = FilterValues( $list ; window::xPt_SelectedItems__CurrentClickedPortalSelectedIDs  ) ;

                        $sl1 = Substitute ( Trim ( Substitute ( LeftValues( $sl ; 1) ; [" " ; "@"]; ["¶";" "] )  )  ;[" ";"¶"]; ["@" ;" "]; ["¶¶¶";"¶"] ;["¶¶";"¶"] ;
                        ["¶¶";"¶"]  ) ;
                        $sl2 = Substitute ( Trim ( Substitute ( RightValues( $sl ; 1) ; [" " ; "@"]; ["¶";" "] )  )  ;[" ";"¶"]; ["@" ;" "]; ["¶¶¶";"¶"] ;["¶¶";"¶"] ;
                        ["¶¶";"¶"]  ) ;
                        $start1 = Position (  Filter( Substitute( Substitute( "TTT" &  $list  & "TTT"; "¶" ; "TTT¶TTT" ) ; "TTT"  &  $sl1  & "TTT" ; "#") ;
                        "#¶"  ) ; "#" ;1; 1) ;
                        $start2 = Position (  Filter( Substitute( Substitute( "TTT" &  $list  & "TTT"; "¶" ; "TTT¶TTT" ) ; "TTT"  &  $sl2  & "TTT" ; "#") ;
                        "#¶"  ) ; "#" ;1; 1) ;
                        $end = Position (  Filter( Substitute(  Substitute( "TTT" &  $list  & "TTT"; "¶" ; "TTT¶TTT" ) ; "TTT"  & $PortalID  & "TTT" ;
                        "#") ; "#¶"  ) ; "#" ;1; 1)
                        ];

                        Let(
                        $fl = Case(
                        $end < $start1; MiddleValues ( $list ; $end ; $start1 -  $end + 1 ) ;
                        $end < $start2; MiddleValues ( $list ; $end ; $start2 -  $end + 1 ) ;
                        $end > $start2; MiddleValues ( $list ; $start2 ; $end - $start2 + 1 ) ;
                        $portalID
                        )  & "¶" & $sl ;

                        Case(
                        WordCount( FilterValues( $fl ; $listclean)  ) = 0 ;  FilterValues( $fl ; $list) ;
                        FilterValues( $fl ; $listclean)

                        //PatternCount( $portalID ; "LCx" )  < 1 ; FilterValues( $fl ; $listclean) ;

                        )

                        )
                        ) ]
        Else If [ $sk = 8 ]
                Set Variable [ $$EnterNote; Value:"Enter" ]
                Set Field By Name [ window::xPt_SelectedItems__CurrentClickedPortalFN; If(
                        PatternCount ( window::xPt_SelectedItems__CurrentClickedPortalSelectedIDs; window::
                        xPt_SelectedItems__CurrentClickedPortalID ) > 0 ;
                        window::xPt_SelectedItems__CurrentClickedPortalSelectedIDs ;
                        window::xPt_SelectedItems__CurrentClickedPortalID
                        ) ]
                Go to Object [ Object Name: Case(
                        GetLayoutObjectAttribute("tab.mainlist 3";"isFrontTabPanel")  =  1 ; "field."  & $$drag.portal  & ".new 3" ;
                        GetLayoutObjectAttribute("tab.mainlist 2";"isFrontTabPanel")  =  1 ; "field."  & $$drag.portal  & ".new 2" ;
                        "field."  & $$drag.portal  & ".new"
                        ) ]
        End If
        Set Variable [ $$ClickTimer; Value:Get(CurrentTime) - 2 ]
        Exit Script [ Result: Let( [
                $$drop.note = "" ;
                $$drag.ID = "" ;
                $$drop.ID = "" ;
                $$drag.portal = "";
                $$drag.note = "";
                $$dragheadername = "";
                $$drop.sidebarID = "" ;
                $$drag.sidebarID = ""  ;
                $$Drag.foldertype = "" ;
                $$Draglist = "" ;
                $$Dontchange = ""
                ] ; "" ) ]
        End If
        #
        If [ window::xPt_SelectedItems__CurrentClickedPortalID = window::xPt_SelectedItems__LastClickedPortalID and
        window::xPt_SelectedItems__CurrentClickedPortal = window::xPt_SelectedItems__LastClickedPortal and
         Get(CurrentTime)    ≤   $$ClickTimer + 1 ]
        #-- Double Click --
        Set Variable [ $$ClickTimer; Value:Get(CurrentTime) - 3 ]
        Perform Script [ “DoubleClicked [ link (optional), $$EnterNote , $$drag.portal]” ]
        Exit Script [  ]
        End If
        #
        #--  First Click--
        Set Field By Name [ window::xPt_SelectedItems__CurrentClickedPortalFN; window::xPt_SelectedItems__CurrentClickedPortalID ]
        If [ window::xPt_SelectedItems__CurrentClickedPortal = "MainList" or
        window::xPt_SelectedItems__CurrentClickedPortal = "CompanyList" ]
        If [ Left(  window::xPt_SelectedItems__CurrentClickedLink  ;1) = "C" ]
                Set Field [ window::xPt_SelectedItems__LastClickedContact; window::xPt_ContactID ]
                Set Field [ window::xPt_ContactID; window::xPt_SelectedItems__CurrentClickedLink ]
                Set Field [ window::xPt_ItemID; "" ]
                Set Field [ window::xPt_Opened_ContactLinkList; "" ]
                Set Field [ window::xPt_SelectedItems_CompanyList; Case(
                        window::xPt_SelectedItems__CurrentClickedPortal = "CompanyList" ;window::xPt_SelectedItems_CompanyList ;
                        window::xPt_SelectedItems__CurrentClickedLink
                        ) ]
        Else
                Set Field [ window::xPt_ItemID; window::xPt_SelectedItems__CurrentClickedLink ]
                If [ not IsEmpty( join.primarycontact::xxIDL_IDNumberInternal )
                        and join.primarycontact::xxIDL_IDNumberInternal ≠ window::xPt_ContactID ]
                        Set Field [ window::xPt_SelectedItems__LastClickedContact; window::xPt_ContactID ]
                        Set Field [ window::xPt_ContactID;   join::xxIDL_LinkPrimaryItem ]
                        Set Field [ window::xPt_SelectedItems_CompanyList; Case(
                        window::xPt_SelectedItems__CurrentClickedPortal = "CompanyList" ;window::xPt_SelectedItems_CompanyList ;
                        window::xPt_SelectedItems__CurrentClickedPortal = "ContactLinkList" ;window::xPt_SelectedItems_CompanyList ;
                        window::xPt_ContactID
                        ) ]
                        Set Field [ window::xPt_SelectedItems_ContactLinkList; Case(
                        PatternCount(  window::xPt_ContactLinkList ; window::xPt_SelectedItems__CurrentClickedLink) > 0 ; window::
                        xPt_SelectedItems__CurrentClickedLink  ;
                        ""
                        ) ]
                        Set Field [ window::xPt_Opened_ContactLinkList; "" ]
                End If
                Set Field By Name [ If(
                        Left( window::xPt_SelectedItems__CurrentClickedLink ;1) = "R" ; "window::xID_EditLinkRegistration"; "" ); window::
                        xPt_SelectedItems__CurrentClickedLink ]
        End If
    #
    #-- Click Reminder Dot --
    Set Field [ window::xID_ReminderID; window::xPt_SelectedItems__CurrentClickedLink ]
    Set Field [ window.reminder::Flag; If( $$drag.portal = "mainlist" and portal.sidebar.folder.selected::xFolder_ItemClearOnClickPref =
    1 ;

        Case(
         window.reminder::Flag = "2"; "2" ;
         window.reminder::Flag = "0"; "0" ;
        "" )
        ) ]
    Set Field [ window::xID_ReminderID; "" ]
End If
If [ window::xPt_MainListLayout = "importemail" ]
    If [ window::xPt_SelectedItems__CurrentClickedPortal = "MainList" ]
        Perform Script [ “ImportEmailWindow [ link ]”; Parameter: window::xPt_SelectedItems__CurrentClickedLink ]
    Else
        If [ Left($$drag.id;1) = "C" ]
            Set Field [ window::xPt_ContactID; $$drag.id ]
            Set Field [ window::xPt_Opened_ContactLinkList; "" ]
        End If
    End If
End If
#
Set Variable [ $$ClickTimer; Value:Get(CurrentTime) ]
Exit Script [  ]
#
#
If [ window::xPt_SelectedItems__CurrentClickedPortalID = window::xPt_SelectedItems__LastClickedPortalID and
    window::xPt_SelectedItems__CurrentClickedPortal = window::xPt_SelectedItems__LastClickedPortal ]
    If [  Get(CurrentTime)    ≤   $$ClickTimer + 1 ]
        #-- Double Click --
        Set Variable [ $$ClickTimer; Value:Get(CurrentTime) - 3 ]
        Perform Script [ “DoubleClicked [ link (optional), $$EnterNote , $$drag.portal]” ]
    Else
        #-- First Click, Same Item, reload contact --
        Set Variable [ $$ClickTimer; Value:Get(CurrentTime) ]
    End If
    Exit Script [  ]
End If
#
#
```

Window: Drag, Drop, and Selctiong: QuickClicked

```
Freeze Window
Set Field [ window::xPt_SelectedItems__CurrentClickedPortalRowMainlist; If( Get ( ActivePortalRowNumber ) < 1 ; window::
    xPt_SelectedItems__CurrentClickedPortalRowMainlist ; Get ( ActivePortalRowNumber )) ]
Set Field [ window::xPt_SelectedItems__LastClickedPortal; window::xPt_SelectedItems__CurrentClickedPortal ]
Set Field [ window::xPt_SelectedItems__LastClickedPortalID; window::xPt_SelectedItems__CurrentClickedPortalID ]
Set Field [ window::xPt_SelectedItems__CurrentClickedPortal; $$drag.portal ]
Set Field [ window::xPt_SelectedItems__CurrentClickedPortalID; $$Drag.PortalID ]
Set Field [ window::xPt_SelectedItems_MainList; $$Drag.PortalID ]
#
#--  First Click--
If [ Left(  $$Drag.PortalID ;1) = "C" ]
    Set Field [ window::xPt_ContactID; $$Drag.PortalID ]
    Set Field [ window::xPt_SelectedItems_CompanyList; $$Drag.PortalID ]
    Set Field [ window::xPt_Opened_ContactLinkList; "" ]
    Set Field [ window::xPt_ItemID; $$Drag.PortalID ]
    Set Window Title [ Current Window; New Title: Case (
        Get(LayoutName)  =  "Folder Options" ; Get(LayoutName);

        Let( [
        $a = If ( PatternCount( Get(LayoutName) ;  "Main Window" ) or PatternCount( Get(LayoutName) ;  "Folder" ) ; portal.sidebar.
        folder.selected::Folder Name  & " - " ; "" ) &
        If ( IsEmpty ( join::link.type.window title ) ; Get ( LayoutName ) ; join::link.type.window title ) ;
        $b = PatternCount ( "¶" & WindowNames & "¶" ; "¶" & $a  & "¶")  ;
        $c = PatternCount ( "¶" & WindowNames ; "¶" & $a )
        ] ;
        Trim ( $a  &  If ( $b = 0  ; "" ;  " " & $c + 1 )  )
        )

        ) ]
    Set Field [ window::xPt_ItemID; "" ]
Else
    Set Field [ window::xPt_ItemID; $$Drag.PortalID ]
    If [ not IsEmpty( join.primarycontact::xxIDL_IDNumberInternal )
        and join.primarycontact::xxIDL_IDNumberInternal ≠ window::xPt_ContactID ]
        Set Field [ window::xPt_ContactID; join.primarycontact::xxIDL_IDNumberInternal ]
        Set Field [ window::xPt_SelectedItems_CompanyList; window::xPt_ContactID ]
        Set Field [ window::xPt_SelectedItems_ContactLinkList; Case(
            PatternCount(  window::xPt_ContactLinkList ; $$Drag.PortalID ) > 0 ; $$Drag.PortalID  ;
            ""
            ) ]
        Set Field [ window::xPt_Opened_ContactLinkList; "" ]
    End If
    Set Window Title [ Current Window; New Title: Case (
        Get(LayoutName)  =  "Folder Options" ; Get(LayoutName);

        Let( [
        $a = If ( PatternCount( Get(LayoutName) ;  "Main Window" ) or PatternCount( Get(LayoutName) ;  "Folder" ) ; portal.sidebar.
        folder.selected::Folder Name  & " - " ; "" ) &
        If ( IsEmpty ( join::link.type.window title ) ; Get ( LayoutName ) ; join::link.type.window title ) ;
        $b = PatternCount ( "¶" & WindowNames & "¶" ; "¶" & $a  & "¶")  ;
        $c = PatternCount ( "¶" & WindowNames ; "¶" & $a )
        ] ;
        Trim ( $a  &  If ( $b = 0  ; "" ;  " " & $c + 1 )  )
        )

        ) ]
End If
#
#-- Click Reminder Dot --
Set Field [ window::xID_ReminderID; $$Drag.PortalID ]
Set Field [ window.reminder::Flag; If(  portal.sidebar.folder.selected::xFolder_ItemClearOnClickPref = 1 ;

    Case(
     window.reminder::Flag = "2"; "2" ;
     window.reminder::Flag = "0"; "0" ;
    "" )
```

```
        ) ]
Set Field [ window::xID_ReminderID; "" ]
#
Set Variable [ $$ClickTimer; Value:Get(CurrentTime) ]
Set Variable [ $$QuickClick; Value:"" ]
#
#
Commit Records/Requests
        [ Skip data entry validation; No dialog ]
Exit Script [  ]
Set Field [ window::xPt_SelectedItems__LastClickedContact; window::xPt_ContactID ]
Refresh Window
```

Window: Drag, Drop, and Selctiong: DoubleClicked [link (optional), $$EnterNote , $$drag.portal]

```
Set Variable [ $link; Value:If( IsEmpty ( Get( ScriptParameter) ) ; window::xPt_SelectedItems__CurrentClickedLink ; Get
    (ScriptParameter) ) ]
Set Variable [ $relatedwindow; Value:window::xxIDL_IDNumberInternal ]
Freeze Window
Set Field [ window::xPt_ItemIDForScript; $link ]
Set Field [ join::xxIDL_IDNumberPortal; $link ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
If [ Left($link;1) = "m" ]
    Set Field [ window::xPt_ItemIDForScript; "" ]
    Set Field [ window::xPt_ItemID; $link ]
    If [ window::xPt_MainListLayout = "importemail" or IsEmpty(item.note::xxIDL_LinkPrimaryItem) ]
        Perform Script [ “ImportEmailWindow [ link ]”; Parameter: $link ]
    Else
        Perform Script [ “WindowNew [ main, folder, folder options, pref, note, newemail, importemail, phonerecord, websites]”;
            Parameter: "newemail" ]
        Set Field [ window::xPt_ItemID; $link ]
        Set Field [ window::xPt_ContactID; item.note::xxIDL_LinkPrimaryItem ]
    End If
Else If [ Left($link;1) = "q" ]
    Go to Related Record [ From table: “item.quote”; External; Using layout: “Quote Entry” (quotes) ]
        [ Show only related records; New window ]
    Perform Script [ “window resize [centered, full]” from file: “quotes”; Parameter: "full" ]
Else If [ Left($link;1) = "o" ]
    Go to Related Record [ From table: “item.order”; External; Using layout: “Order” (Order) ]
        [ Show only related records; New window ]
Else If [ Left($link;1) = "p" ]
    Go to Related Record [ From table: “item.PO”; External; Using layout: “Purchase Order” (PO) ]
        [ Show only related records; New window ]
Else If [ Left($link;1) = "D" and $$EnterNote  ≠ "key" ]
    Go to Related Record [ From table: “item.note”; Using layout: “data.note” (item.note) ]
        [ Show only related records; New window ]
    If [ Get ( WindowName )  = "~temp~" ]
        Go to Field [ item.note::File Container ]
            [ Select/perform ]
        Close Window [ Current Window ]
    End If
Else If [ Left( $link ;1) = "C" ]
    Set Field [ window::xPt_ItemIDForScript; "" ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    New Window [ Height: 1; Width: 1; Top: -350 ]
    Go to Layout [ “Contact Card” (window) ]
    Show/Hide Status Area
        [ Hide ]
    New Record/Request
    Set Field [ window::xPt_ContactID; $link ]
    Set Field [ window::xID_RelatedWindow; $relatedwindow ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Perform Script [ “ItemTab” ]
    Move/Resize Window [ Current Window; Height: 764; Width: 520; Top: 158; Left: 712 ]
    Allow User Abort [ Off ]
    Pause/Resume Script [ Duration (seconds): .7 ]
Else
    Set Field [ window::xPt_ItemIDForScript; "" ]
    Set Variable [ $AC; Value:window::xPt_ContactID ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    New Window [ Height: 1; Width: 1; Top: -300 ]
    Go to Layout [ “Item Card” (window) ]
    Show/Hide Status Area
        [ Hide ]
    New Record/Request
    Set Field [ window::xPt_ItemID; $link ]
    Set Field [ window::xID_RelatedWindow; $relatedwindow ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Set Field [ window::xPt_ContactID; $ac ]
    If [ 1 = 2 //not IsEmpty( item.note::xxIDL_LinkPrimaryItem ) and  Left(LeftWords(item.note::xxIDL_LinkPrimaryItem ;1) ; 1) = "c" ]
        Set Field [ window::xPt_ContactID; item.note::xxIDL_LinkPrimaryItem ]
    End If
    Go to Object [ Object Name: "tab.item."  & join::link.type.tab ]
    Perform Script [ “ItemTab” ]
    Move/Resize Window [ Current Window; Height: Case(
        join::link.type.tab = "note"; 300 + Let( a =  PatternCount (item.note::Note ; "¶" ); Case (a > 25; 25*14; a > 5; a * 14; 0) ) ;
        300
        ); Width: 507; Top: 467; Left: 647 ]
    Allow User Abort [ Off ]
    Pause/Resume Script [ Duration (seconds): .7 ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
End If
Set Field [ window::xPt_ItemIDForScript; "" ]
Exit Script [  ]
New Window [ Height: 635; Width: 476; Top: 304; Left: 642 ]
```

Window: Drag, Drop, and Selctiong: OpenItem [$$drag.id, $$drag.portal]

```
Freeze Window
If [ Left( $$drag.id ; 1)  ≠ "L" ]
        Set Field By Name [ "window::xPT_Opened_"  & $$drag.portal; Let( [
                $O = Evaluate(  "window::xPT_Opened_"  & $$drag.portal ) ;
                $cl = List( Evaluate( "portal."  & $$drag.portal  & "::xxIDL_IDNumberInternal") ) ;
                $L =  If( PatternCount( FilterValues( $cl ; $O)  ; $O ) > 0 ; $O ; "")
                ] ;

                Case(
                PatternCount( "¶" & $L & "¶" ; "¶"  & $$drag.id  & "¶") > 0 ;
                Substitute("¶" &  $L  & "¶"; ["¶"  & $$drag.id & "¶" ; "¶" ] ; ["¶¶" ; "¶" ] ; ["¶¶" ; "¶" ]; ["¶¶" ; "¶" ]) ;

                $$drag.id  & "¶"  &  $L
                )
                ) ]
End If
Commit Records/Requests
        [ Skip data entry validation; No dialog ]
Exit Script [ Result: Let( [
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$Dontchange = ""
        ] ; "" ) ]
```

Window: Drag, Drop, and Selctiong: SidebarItemSelect [link]

```
Set Field [ window::xPt_ItemIDForScript; "" ]
Set Variable [ $a; Value:Let( [
        $sp =  Get(ScriptParameter);
        $link =  MiddleWords(  $sp; 1 ; 1 ) ;
        $note = MiddleWords( $sp; 2 ; 1 ) ;
        $mk = If( $$time + 1 > Get(CurrentTime) and $$SelectID = $link ; 1 ; Get ( ActiveModifierKeys ) )  //double click timer
        ];"")

        /*
        Shift = 1,
        Caps Lock = 2,
        Ctrl (Windows) and Control (Mac OS) = 4,
        Alt (Windows) and Option (Mac OS) = 8,
        ⌘ (Mac OS) = 16
        */ ]
If [ Left($link;1) = "C" ]
        Set Field [ window::xPt_ContactID; $link ]
        Set Field [ window::xPt_ItemID; "" ]
        Set Field [ window::xPt_Opened_ContactLinkList; "" ]
        Set Field [ window::xPt_SelectedItems_MainList; If( PatternCount ( window::xPt_MainListIDlist ; $link ) > 0 ; $link; window::
                xPt_SelectedItems_MainList  ) ]
Else If [ Left($link;1) = "D" ]
Else
        Set Field [ window::xPt_ItemID; $link ]
        Set Field [ window::xPt_ContactID;   join::xxIDL_LinkPrimaryItem ]
        Set Field [ window::xPt_Opened_ContactLinkList; "" ]
        Set Field [ window::xPt_SelectedItems_MainList; Case(
                PatternCount ( window::xPt_MainListIDlist ; $link ) > 0 ; $link;
                 window::xPt_SelectedItems_MainList = window::xPt_ContactID; window::xPt_SelectedItems_MainList ;
                 ""
                ) ]
End If
Set Field [ window::xPt_ItemIDForScript; "" ]
Perform Script [ “ItemTab” ]
Set Field [ window::xPt_SelectedItems__CurrentClickedPortal; "" ]
Commit Records/Requests
        [ Skip data entry validation; No dialog ]
```

Window: Drag, Drop, and Selctiong: Linking: LinkItems [$$drop.id[1], $$draglist[2]] / NewLinks [$$dr…

```
#creates new link for $$drop.id if no $$draglist, only one at a time though
Freeze Window
Set Variable [ $a; Value:Let( [
    $$drop.id = If( IsEmpty( window::xSys_NewItem[1] ) ; $$drop.id; window::xSys_NewItem[1] ) ;
    $$DragList = If( IsEmpty( window::xSys_NewItem[1] ) ; $$DragList; window::xSys_NewItem[2] )
    ] ; ""  ) ]
#
Set Field [ window::xPt_ItemIDForScript; $$drop.id ]
Set Field [ join::xxIDL_IDNumberPortal; $$drop.id ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
#
Set Variable [ $end; Value:Let(   [
    $LL = $$draglist  ;
    $count = 0  ;
    $lltemp = window::xPt_ItemIDForScript
    ];
    WordCount($LL)  ) ]
Loop
    Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
    Set Variable [ $link; Value: MiddleWords($LL; $count ; 1 ) ]
    Set Field [ window::xPt_ItemIDForScript; $link ]
    If [ Left( window::xPt_ItemIDForScript ; 1)  ≠ "L" ]
        Set Field [ join::xxIDL_IDNumberPortal; $link ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
    Else
        Set Variable [ $link; Value:join::xxIDL_IDNumberInternal ]
    End If
    If [ not IsEmpty( $$drop.id) ]
        Set Field [ window::xPt_ItemIDForScript; "LCx"  &  $link  & "LPx"   & $$drop.id ]
        Set Field [ join::xxIDL_IDNumberPortal; window::xPt_ItemIDForScript ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
        Set Field [ window::xPt_ItemIDForScript; "LCx"  & $$drop.id   & "LPx"   & $link ]
        Set Field [ join::xxIDL_IDNumberPortal; window::xPt_ItemIDForScript ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
    End If
End Loop
Set Field [ window::xPt_ItemIDForScript; $lltemp ]
#
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
If [ IsEmpty( window::xSys_NewItem ) ]
    Exit Script [  ]
Else
    Set Field [ window::xSys_NewItem; "" ]
    Set Field [ window::xSys_NewItem[2]; "" ]
    Exit Script [ Result: Let( [
        $$drop.note = "" ;
        $$drag.ID = "" ;
        $$drag.PortalID = "" ;
        $$drop.ID = "" ;
        $$drag.portal = "";
        $$drag.note = "";
        $$dragheadername = "";
        $$drop.sidebarID = "" ;
        $$drag.sidebarID = ""  ;
        $$Drag.foldertype = "" ;
        $$Draglist = "" ;
        $$DraglistPortalID = "" ;
        $$Dontchange = ""
        ] ; "" ) ]
End If
```

Window: Drag, Drop, and Selctiong: Linking: LinkSetPrimary [PrimaryContact, link]

```
Freeze Window
Set Error Capture [ Off ]
Set Variable [ $PC; Value:MiddleWords ( Get(ScriptParameter) ; 1; 1) ]
Set Variable [ $link; Value:MiddleWords ( Get(ScriptParameter) ; 2; 1) ]
If [ Left( $PC ;1) = "C" ]
    Set Field [ window::xPt_ItemIDForScript; $link ]
    Set Field By Name [ join::xxIDL_FNLinkPrimaryItem; $PC ]
    Set Field [ window::xPt_ItemIDForScript; "" ]
End If
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Perform Script [ “UpdateLinks [ links ] ”; Parameter: $link ]
Exit Script [  ]
```

Window: Drag, Drop, and Selctiong: Linking: LinkRemove [$$drop.id, $$draglist, cleans]

```
Freeze Window
Set Variable [ $end; Value:Let(   [
    $drop = $$drop.id ;
    $LL = $$DragList  ;
    $lltemp = window::xPt_ItemIDForScript;
    $count = 0
    ];
    WordCount($LL)  ) ]
Loop
    Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
    Set Variable [ $link; Value: MiddleWords($LL; $count ; 1 ) ]
    Set Variable [ $item; Value:If(
        Left( $link ; 1 )   ≠   "L" ; $drop ;
        Middle ( $link ; Position ( $link ; "LPx" ; 1; 1) + 3; 20)
         ) ]
    Set Variable [ $link; Value:If(
        Left( $link ; 1 )   ≠   "L" ; $link ;
        Middle ( $link; 4 ; Position ( $link ; "LPx" ; 1; 1) - 4 )
         ) ]
    If [ WordCount ( $link ) = 1 and WordCount ($item) = 1 ]
        Set Field [ window::xPt_ItemIDForScript; "LCx"  &  $link  & "LPx"   & $item ]
        If [ not IsEmpty(join::xxIDL_IDNumberPortal) ]
            Set Variable [ $PCItem; Value:join.primarycontact::Name ]
            Set Field [ join::xxIDL_IDNumberPortal; "Delete" ]
        End If
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
        Set Field [ window::xPt_ItemIDForScript; "LCx"  &  $item  & "LPx"   & $link ]
        If [ not IsEmpty(join::xxIDL_IDNumberPortal) ]
            Set Variable [ $PCLink; Value:join.primarycontact::Name ]
            Set Field [ join::xxIDL_IDNumberPortal; "Delete" ]
        End If
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
        #
        Set Field [ window::xPt_ItemIDForScript; $link ]
        Set Field [ join::xxIDL_IDNumberPortal; $link ]
        Perform Script [ “Move/Copy/ClearToAnyFolder [ $$drop.ID [1], $$DragList [2] cleans, g(sp) = from folders ] ”; Parameter: Let( [
            $$Drop.Id = If ( $$DeleteKeyItem = "Move Orphaned to Trash" ; $$TrashID; "" ) ;
            $$draglist = If( join::xxIDL_Orphaned; $link  ; "" )
            ] ; "") ]
        Set Field By Name [ Case(
            join::link.type.table  ≠  "note"  ; "" ;
            PatternCount( Evaluate( join::xxIDL_FNLinkPrimaryItem ) ; $item ) = 0 ; "" ;
            join::xxIDL_LinkNotValid ; "" ;
            "item.note::Note"
            )

            /*  If( join::link.type.table = "note" and
            PatternCount( Evaluate( join::xxIDL_FNLinkPrimaryItem ) ; $item ) > 0
            and not join::xxIDL_LinkNotValid
            ; "item.note::Note"; "" )  */; item.note::Note & "  (Original note contact: "  & $PClink & ")" ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
        #
        Set Field [ window::xPt_ItemIDForScript; $item ]
        Set Field [ join::xxIDL_IDNumberPortal; $item ]
        Perform Script [ “Move/Copy/ClearToAnyFolder [ $$drop.ID [1], $$DragList [2] cleans, g(sp) = from folders ] ”; Parameter: Let( [
            $$Drop.Id = If ( $$DeleteKeyItem = "Move Orphaned to Trash" ; $$TrashID; "" ) ;
            $$draglist = If( join::xxIDL_Orphaned; $item  ; "" )
            ] ; "") ]
        Set Field By Name [
            Case(
            join::link.type.table  ≠  "note"  ; "" ;
            PatternCount( Evaluate( join::xxIDL_FNLinkPrimaryItem ) ; $link ) = 0 ; "" ;
            join::xxIDL_LinkNotValid ; "" ;
            "item.note::Note"
            )


            /*  If(
            join::link.type.table = "note" and PatternCount( Evaluate( join::xxIDL_FNLinkPrimaryItem ) ; $link ) > 0  and  not join::
            xxIDL_LinkNotValid;  "item.note::Note" ;
             "" )
            */; item.note::Note & "  (Original note contact: " & $PCItem & ")" ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
    End If
End Loop
#
Set Field [ window::xPt_ItemIDForScript; $lltemp ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Exit Script [ Result: Let( [$$drop.id = $drop ; $$draglist = $LL ]; "" ) ]
#
```

Window: Drag, Drop, and Selctiong: Linking: UpdateLinks [links]

```
Freeze Window
Set Variable [ $lltemp; Value: window::xPt_ItemIDForScript ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Set Variable [ $link; Value:Get(ScriptParameter) ]
If [ Left($link; 1 )  = "F" ]
    Set Field [ window::xPt_ItemIDForScript; LeftWords( $link ; 1) ]
    Set Variable [ $sort; Value:item.folder::xFSelected_SortMainlistChoice ]
    Set Field [ window::xPt_ItemIDForScript; item.folder::xxIDL_LinkListCalc ]
    Set Variable [ $cl; Value:List( join::xxIDL_IDNumberInternal ) ]
    Set Variable [ $LL; Value:Case(
        $sort  = "Company" ; List( join::link.sort.field.modifedtIDitem )  ;
        $sort  = "contact" ;   List( join::link.sort.field.modifedtIDitem )  ;
        $sort  = "reminder date" ; List( join::link.sort.field.modifedtIDreminder )  ;
        $sort  = "reminder note" ; List( join::link.sort.field.modifedtIDreminder )  ;
        $sort  = "reminder flag" ; List( join::link.sort.field.modifedtIDreminder )  ;
        ""
        ) ]
    Set Field [ window::xPt_ItemIDForScript; "" ]
Else If [ IsEmpty( $link) ]
    Set Field [ window::xPt_ItemIDForScript; window::xPt_ContactID & "¶" & window::xPt_ItemIDcalc ]
    Set Variable [ $LL; Value:List( join::xxIDL_IDNumberInternal ) ]
    Set Field [ window::xPt_ItemIDForScript; "" ]
Else
    Set Field [ window::xPt_ItemIDForScript; $link ]
    Set Variable [ $LL; Value:List( join::xxIDL_IDNumberInternal ) ]
    Set Field [ window::xPt_ItemIDForScript; "" ]
End If
Set Variable [ $a; Value:Let(  [
    $end =WordCount( $LL )
    ] ; ""
    ) ]
Loop
    Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
    Set Field [ window::xPt_ItemIDForScript; MiddleWords( $LL; $count; 1) ]
    Set Field [ join::link.sort.field.modifedtimestamp; Get(CurrentTimeStamp) ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
End Loop
Set Field [ window::xPt_ItemIDForScript; $lltemp ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Exit Script [  ]
#
Set Field [ join::link.sort.field.remindernote; item.reminder::Note & "1" & join::xxIDL_IDNumberPortal ]
Set Field [ join::link.sort.field.reminderdate desc;  item.reminder::xxCF_DateTimeSort  & "1" & join::xxIDL_IDNumberPortal ]
Set Field [ join::link.sort.field.reminderflag; item.reminder::Flag & "1" & join::xxIDL_IDNumberPortal ]
Set Field [ join::link.sort.field.contact; join::link.primary.contact name ]
Set Field [ join::link.sort.field.company; join::link.primary.company name ]
Show Custom Dialog [ Message: $end; Buttons: “OK”, “Cancel” ]
Show Custom Dialog [ Message: WordCount(window::xPt_ItemIDForScript ); Buttons: “OK”, “Cancel” ]
```

Window: Drag, Drop, and Selctiong: Linking: Delete [links]

```
Freeze Window
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Set Variable [ $a; Value:Let(  [
    $count = 0 ;
    $DeleteList =  Get(ScriptParameter) ;
    $lltemp = window::xPt_ItemIDForScript;
    $LL = Get(ScriptParameter) ;
    $end =WordCount( $LL )
    ] ; ""
    ) ]
Loop
    Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
    Set Variable [ $link; Value:MiddleWords( $LL; $count; 1) ]
    Set Field [ window::xPt_ItemIDForScript; $link ]
    Set Variable [ $a; Value:Let( [
        $count2 = 0 ;
        $LL2 = join::xxIDL_LinkListTotal ;
        $end2 = WordCount(  $LL2 )
        ] ; "") ]
    Loop
        Exit Loop If [ Let( $count2 = $count2 + 1 ; $count2)   >  $end2 or $end2 < 1 ]
        Set Variable [ $link2; Value:MiddleWords( $LL2; $count2; 1) ]
        Set Variable [ $PClink; Value:join.primarycontact::Name ]
        Set Field [ window::xPt_ItemIDForScript; $link2 ]
        Set Field By Name [ If(
            not IsEmpty(join::xxIDL_IDNumberPortal ); "" ;
            "join::xxIDL_IDNumberPortal"
            ); $link2 ]
        If [ Case(
            PatternCount( "¶" & $$DeleteKeyOrphaned  & "¶" ; "¶" &  join::link.type.detailed  & "¶" ) =  0 ; "" ;
            WordCount(  join::xxIDL_LinkListTotal ) > 1 ; "" ;
            WordCount(  join::xxIDL_LinkListTotal ) = 0 ; 1 ;
            LeftWords( join::xxIDL_LinkListTotal ;1) = $link ; 1 ;
            join::xxIDL_LinkNotValid ;1 ;
            "" ) ]
            Set Variable [ $deletelist; Value:$deletelist   & "¶" &  $link2   & "¶"  &
                 "LCx"  &  $link  & "LPx"   & $link2 & "¶"  &
                "LCx"  &  $link2  & "LPx"   & $link ]
        Else
            Set Field By Name [ Case (
                join::link.type.table  ≠  "note" ;"" ;
                LeftWords( join::xxIDL_FNLinkPrimaryItem ;1)   ≠  $link ; "" ;
                "item.note::Note" ); item.note::Note & "  (Original note contact: "  & $PClink & ")" ]
            Set Variable [ $deletelist; Value:$deletelist   & "¶"  &
                "LCx"  &  $link  & "LPx"   & $link2 ]
            Commit Records/Requests
                [ Skip data entry validation; No dialog ]
            Set Field [ window::xPt_ItemIDForScript; "LCx"  &  $link2  & "LPx"   & $link ]
            Set Field [ join::xxIDL_IDNumberPortal; "Delete" ]
        End If
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
    End Loop
End Loop
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
#
Set Field [ window::xPt_ItemIDForScript; "delete" ]
Set Field [ join::xxIDL_IDNumberPortal; "delete" ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Set Variable [ $deletelist; Value: $deletelist   & "¶" &  "delete" ]
Set Field [ window::xPt_ItemIDForScript; $deletelist ]
```

```
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Go to Related Record [ From table: “join”; Using layout: “data.join” (join) ]
    [ Show only related records; New window ]
If [ Get ( LayoutName ) = "data.join" ]
    Delete All Records
        [ No dialog ]
End If
If [ Get ( LayoutName ) = "data.join" or Get(WindowName) = "~temp~" ]
    Close Window [ Current Window ]
End If
#
Set Field [ window::xPt_ItemIDForScript; $lltemp ]
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
Exit Script [  ]
Set Field By Name [ Case(
    Left( $link2; 1 ) = "c" ; "" ;
    join::xxIDL_FNLinkPrimaryItem ); Let( a =  Evaluate ( join::xxIDL_FNLinkPrimaryItem );
    Case(
    PatternCount("¶" &  a ; "¶" & $link ) > 0; "" ; a  )
    ) ]
```

duplicate folders

```
Sort Records [ Specified Sort Order: item.folder::xxIDL_IDNumberInternal; ascending ]
    [ Restore; No dialog ]
Go to Record/Request/Page
    [ First ]
Loop
    Duplicate Record/Request
    Set Field [ item.folder::xxIDL_IDNumberInternal;  "F" &
        Right("00" & Day(Get(CurrentDate));2)  &
        Right("00" & Month(Get(CurrentDate));2) &
        Right(Year(Get(CurrentDate));2)  &
        Right("00000" & GetAsText(Get(RecordID));5) &
        Right("0000" & GetAsText(Int(9999*Random));4) ]
    Set Field [ item.folder::xFolder_Template; "" ]
    Set Field [ item.folder::xxIDL_Owner; "KMC" ]
    Set Field [ item.folder::xxIDL_LinkList; "" ]
    Go to Record/Request/Page
        [ Next; Exit after last ]
End Loop
```

create folder join links

```
#
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
#
Set Variable [ $end; Value:Let(   [
    $LL = List(portal.sidebar.folder.all::xxIDL_IDNumberInternal) ;
    $count = 0
    ];
    WordCount($LL)  ) ]
Show Custom Dialog [ Message: $end & "¶" & $LL; Buttons: “OK”, “Cancel” ]
Loop
    Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
    Set Variable [ $link; Value: MiddleWords($LL; $count ; 1 ) ]
    Set Field [ window::xPt_ItemIDForScript; $link ]
    Set Field [ join::xxIDL_IDNumberPortal; $link ]
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
End Loop
Set Field [ window::xPt_ItemIDForScript; "" ]
```

Paste Contact Info

```
Set Field [ UserNotes::Note; UserNotes::Note & "¶¶" & window.contact::xSys_ContactInfo ]
```

Copy Contact Info to Clipboard

```
Freeze Window
Set Variable [ $temp; Value:window.contact::Title ]
Set Field [ window.contact::Title; window.contact::xSys_ContactInfo ]
Copy [ window.contact::Department ]
        [ Select ]
Set Field [ window.contact::Title; $temp ]
Commit Records/Requests
        [ Skip data entry validation; No dialog ]
```

Copy Customer ID

```
Freeze Window
Set Variable [ $temp; Value:window.contact::Department ]
Set Field [ window.contact::Department; window.contact::xxIDL_IDNumberInternal ]
Copy [ window.contact::Department ]
        [ Select ]
Set Field [ window.contact::Department; $temp ]
Commit Records/Requests
        [ Skip data entry validation; No dialog ]
```

New Script Copy

```
Set Variable [ $end; Value:Let(   [
        $count = 0
        ];
        80  ) ]
Loop
        Exit Loop If [ Let( $count = $count + 1 ; $count)   >  $end or $end < 1 ]
        Open URL [ "translate.google.com" ]
                [ No dialog ]
        Pause/Resume Script [ Duration (seconds): .2 ]
        Open URL [ "sex.com" ]
                [ No dialog ]
        Pause/Resume Script [ Duration (seconds): .2 ]
End Loop
```

UserWepPages [open, type, openURL, account, password]

```
If [ MiddleWords( Get(ScriptParameter) ; 1 ; 1)  = "type" ]
    Set Field [ window::xID_UserWebType; MiddleWords( Get(ScriptParameter) ; 2 ; 1) ]
Else If [ Get(ScriptParameter) = "openURL" ]
    If [ Get ( ActiveModifierKeys ) = 4 or Get ( ActiveModifierKeys ) = 8 or Get ( ActiveModifierKeys ) = 18 ]
        Paste [ UserWebSites::WebImage ]
            [ Select; No style ]
    Else If [ Get ( ActiveModifierKeys ) = 1 ]
        Show Custom Dialog [ Message: "Remove Image?"; Buttons: “OK”, “Cancel” ]
        If [ Get ( LastMessageChoice ) = 1 ]
            Set Field [ UserWebSites::WebImage; "" ]
        End If
    Else
        Set Variable [ $a; Value: UserWebSites::WebURL ]
        Copy [ UserWebSites::WebAccount ]
            [ Select ]
        Commit Records/Requests
            [ Skip data entry validation; No dialog ]
        Pause/Resume Script [ Duration (seconds): .6 ]
        Copy [ UserWebSites::WebPassword ]
            [ Select ]
        Set Field [ UserWebSites::WebLastAccess; Get(CurrentDate) ]
        Open URL [ $a ]
            [ No dialog ]
    End If
Else If [ Get(ScriptParameter) = "account" ]
    Set Field [ UserWebSites::WebLastAccess; Get(CurrentDate) ]
    Copy [ UserWebSites::WebAccount ]
        [ Select ]
    Send Event [ open application; Application Name: “Safari” ]
        [ Wait for completion; Bring application to foreground ]
Else If [ Get(ScriptParameter) = "password" ]
    Set Field [ UserWebSites::WebLastAccess; Get(CurrentDate) ]
    Copy [ UserWebSites::WebPassword ]
        [ Select ]
    Send Event [ open application; Application Name: “Safari” ]
        [ Wait for completion; Bring application to foreground ]
End If
Commit Records/Requests
    [ Skip data entry validation; No dialog ]
```

export jenna report, just contacts

```
Freeze Window
Go to Layout [ “data.contact” (item.contact) ]
Set Field [ window::sys.GeneralDialogAnswers; 90 ]
Show Custom Dialog [ Message: "report options..."; Buttons: “OK”, “Cancel”; Input #1: window::sys.GeneralDialogAnswers, "days
    prior..." ]
Set Variable [ $f; Value:"file:../../Desktop/Report.xls" ]
Set Variable [ $fd; Value:Get(CurrentDate) - window::sys.GeneralDialogAnswers ]
Perform Find [ Specified Find Requests: Find Records; Criteria: item.contact::Call Last Connect: “>$fd” AND item.contact::Call Owner: “K”
    Omit Records; Criteria: item.contact::Call Category: “res”
    Omit Records; Criteria: item.contact::Call Category: “arch” ]
    [ Restore ]
Export Records [ File Name: “$f”; Use field names as column names; Character Set: “Unicode (UTF-16)”; Field Order: item.contact::Call
Category
    item.contact::Company Name
    item.contact::Name
    item.contact::Title
    item.contact::Department
    item.contact::Email
    item.contact::Region
    item.contact::Phone Area Code
    item.contact::Phone
    item.contact::Ship Address
    item.contact::Ship City
    item.contact::Ship State
    item.contact::Ship Zip ]
    [ No dialog ]
Go to Layout [ original layout ]
```

export jenna report from notes

```
Freeze Window
Go to Layout [ “data.note” (item.note) ]
Set Field [ window::sys.GeneralDialogAnswers; 90 ]
Show Custom Dialog [ Message: "report options..."; Buttons: “OK”, “Cancel”; Input #1: window::sys.GeneralDialogAnswers, "days
    prior..." ]
Set Variable [ $f; Value:"file:../../Desktop/Report.xls" ]
Set Variable [ $fd; Value:Get(CurrentDate) - window::sys.GeneralDialogAnswers ]
Perform Find [ Specified Find Requests: Find Records; Criteria: item.note::Date: “>$fd” AND item.note::xxIDL_Owner: “K” AND item.note::
xSys_Type: “com”
    Find Records; Criteria: item.note::Date: “>$fd” AND item.note::xxIDL_Owner: “K” AND item.note::xSys_Type: “note”
    Find Records; Criteria: item.note::Date: “>$fd” AND item.note::xxIDL_Owner: “K” AND item.note::xSys_Type: “miss” ]
    [ Restore ]
Sort Records [ Specified Sort Order: item.note.primarycontact::Company Name; ascending
    item.note.primarycontact::Name; ascending
    item.note::Date; descending ]
    [ Restore; No dialog ]
Export Records [ Character Set: “Unicode (UTF-16)”; Field Order: item.note.primarycontact::Company Name
    item.note.primarycontact::Name
    item.note.primarycontact::Department
    item.note.primarycontact::Email
    item.note.primarycontact::Phone Area Code
    item.note.primarycontact::Phone
    item.note.primarycontact::Ship City
    item.note.primarycontact::Ship State
    item.note::xSys_Type
    item.note::Note ]
    [ No dialog ]
Go to Layout [ original layout ]
```

ZZNew [[1] type, [2] link to, [3] primaryID, [4] Portal, [5] Note, [6] Ref ID] 2

```
Freeze Window
Set Variable [ $lItemp; Value:window::xPt_ItemIDForScript ]
Set Variable [ $$NewItem; Value:Case(
        IsEmpty( Trim ( window::xSys_NewMenuChoice )  ) ; $$NewItem[1] ;
        Trim ( window::xSys_NewMenuChoice )
        ) ]
#[1] TYPE, [2] LINK TO [3] PRIMARY ID [4] PORTAL NAME [5] NOTE [6] Reference ID [7] Reminder Date [8] Reminder Note, [9] Folder
        ID
#
Set Field [ window::xSys_NewMenuChoice; " " ]
Commit Records/Requests
        [ Skip data entry validation; No dialog ]
#
Loop
        Exit Loop If [ IsEmpty ( Trim( $$NewItem ) ) ]
        #Set Primary Contact
        Set Field [ window::xPt_ItemIDForScript; $$NewItem[3] ]
        Set Variable [ $PC; Value:Evaluate( join::xxIDL_FNLinkPrimaryItem ) ]
        #
        Set Field [ window::xPt_ItemIDForScript; Let( $link =
                Case(
                $$NewItem[1] = "New Opportunity" ; "AO";
                $$NewItem[1] = "New Prospect" ; "AP";
                $$NewItem[1] = "New To Do" ; "AT";
                $$NewItem[1] = "New Call" ; "AC";
                $$NewItem[1] = "New Appointment" ; "AS";
                $$NewItem[1] = "New Scheduled" ; "AS";

                $$NewItem[1] = "New Registration" ; "R";

                $$NewItem[1] = "New File" ; "D";

                $$NewItem[1] = "New Quote" ; "Q";
                $$NewItem[1] = "New Order" ; "O";
                $$NewItem[1] = "New PO" ; "P";

                $$NewItem[1] = "New Contact" ; "C";
                $$NewItem[1] = "New Contact Same Company" ; "C";
                $$NewItem[1] = "Same Contact New Company" ; "C";
                $$NewItem[1] = "New Contact Same Company As Selected" ; "C";

                $$NewItem[1] = "New Email Received" ; "M";
                $$NewItem[1] = "New Email" ; "M";

                $$NewItem[1] = "New Reg. History" ; "V";
                $$NewItem[1] = "New Print" ; "Z" ;
                "N"
                )  &
                Right("00" & Day(Get(CurrentDate));2)  &
                Right("00" & Month(Get(CurrentDate));2) &
                Right(Year(Get(CurrentDate));2)  &
                Right("00000" & GetAsText(Get(RecordID));5) &
                Right("0000" & GetAsText(Int(9999*Random));4)
                ; $link ) ]
        Set Field [ join::xxIDL_IDNumberPortal; $link ]
        Commit Records/Requests
                [ Skip data entry validation; No dialog ]
        Go to Related Record [ From table: “join”; Using layout: “data.join” (join) ]
                [ Show only related records; New window ]
        Set Field By Name [ join::xxIDL_FNIDNumberInternal; $link ]
        Set Field By Name [ If (
                Left( $link; 1)  ≠ "C"; join::xxIDL_FNLinkPrimaryItem ;
                ""); $PC ]
        Commit Records/Requests
                [ Skip data entry validation; No dialog ]
        If [ $$NewItem[1] = "New File" ]
                Insert File [ item.note::File Container ]
                        [ Reference ]
                If [ Get ( LastError )  ≠ 0 ]
```

```
            Delete Record/Request
                [ No dialog ]
            Close Window [ Current Window ]
            Exit Loop If [ 1 ]
        End If
    End If
    Close Window [ Current Window ]
    Set Field [ window::xID_ReminderID; $link ]
    Set Field [ window.reminder::Date; GetAsDate(  $$NewItem[7] ) ]
    Set Field [ window.reminder::Note; $$NewItem[8] ]
    #
    #link
    Perform Script [ “LinkItems [$$drop.id[1], $$draglist[2] ] / NewLinks [ $$drop.id [1] ]”; Parameter: Let( [$$drop.id = $link; $$draglist  =
        $$NewItem[2]] ; "" ) ]
    Set Field [ window::xPt_ItemIDForScript; $link ]
    #move to inbox ($$draglist becomes $link or "")
    Perform Script [ “Move/Copy/ClearToAnyFolder [ $$drop.ID [1], $$DragList [2] cleans, g(sp) = from folders ] ”; Parameter: Let( [

        $$drop.ID =
        Case(
        not IsEmpty( $$NewItem[9]) ; $$NewItem[9] ;
        PatternCount( window.user::xPrefUser_CopyToInboxWhenCreated; join::link.type.detailed ) > 0 ; $$inboxID ;
        "" ) ;

        $$DragList =
        If( PatternCount( window.user::xPrefUser_CopyToInboxWhenCreated; join::link.type.detailed ) > 0 or not IsEmpty( $$NewItem
        [9] );$link ;
        "" )

         ] ; "") ]
    Set Field [ window::xPt_ItemIDForScript; $link ]
    #item specific
    Exit Loop If [ $NewItemType = "New File" ]
    If [ $$NewItem[1] = "New Quote" ]
        Set Field [ item.quote.revision::System rev Quote date; Get(CurrentDate) ]
    Else If [ $$NewItem[1] = "New Contact" or
        $$NewItem[1] = "Same Contact New Company" or
        $$NewItem[1] = "New Contact Same Company" ]
        If [ $$NewItem[1] = "Same Contact New Company" ]
            Perform Script [ “SameContactNewCompany [$ac (old), link (new)]”; Parameter:  $$NewItem[6]  & " "  & $link ]
        Else If [ $$NewItem[1] = "New Contact Same Company" ]
            Perform Script [ “NewContactSameCompany [$ac (old), link (new)]”; Parameter: Case(
                $$NewItem[1] = "New Contact Same Company" ; $$NewItem[6];
                $$NewItem[1] = "New Contact Same Company As Selected"  and Left( window::
                xPt_SelectedItems__CurrentClickedLink ; 1 ) = "C" ; window::xPt_SelectedItems__CurrentClickedLink;
                "")
                 & " "  & $link ]
        Else If [ $$NewItem[1] = "New Contact" ]
        End If
        If [ window::xPt_MainListLayout  =  "importemail" or
            window::xPt_MainListLayout  = "ResearchCard" ]
            Set Field [ window::xPt_ContactID; $link ]
        Else
            Perform Script [ “DoubleClicked [ link (optional), $$EnterNote , $$drag.portal]”; Parameter: $link ]
        End If
        Exit Loop If [ 1 ]
    Else If [ $$NewItem[1] = "New Email" ]
        Perform Script [ “WindowNew [ main, folder, folder options, pref, note, newemail, importemail, phonerecord, websites]”;
            Parameter: "newemail" ]
        Set Field [ window::xPt_ContactID; $PC ]
        Set Field [ window::xPt_ItemID; $link ]
        Set Field [ item.note::Email Recipient List; window.contact::Email ]
        Set Field [ item.note::Email Sender Email; window.user::user.email ]
        Set Field [ item.note::Email Sender Name; window.user::user.name ]
        Set Field [ item.note::Email Status; "Unsent" ]
        Go to Object [ Object Name: "field.emailsubject" ]
        Exit Loop If [ 1 ]
    Else If [ $$NewItem[1] = "New Reg. History"   or
        $$NewItem[1] = "New Print" ]
        Set Field [ item.note::Note; $$NewItem[5] ]
    Else If [ Left($link ;1 )  = "A" ]
        Set Field [ item.action::Note; Case(
            Trim ( $$NewItem[5] )  ≠ Trim( $$NewItem[1] ) ; $$NewItem[5] ;
            ""
            ) ]
        Perform Script [ “DoubleClicked [ link (optional), $$EnterNote , $$drag.portal]”; Parameter: $link ]
    End If
    If [ Get(LayoutName)  =  "Email Import" ]
        Set Field [ window::xID_EmailimportRegistration; Case(
            Left( $link ;1 ) = "R" ; $Link ; window::xID_EditLinkRegistration ) ]
        Set Field By Name [ Case(
            Left( $link ;1 ) = "R" ; "window.registration::sys.lookup paste text" ; "" ); item.note::Note ]
        Go to Object [ Object Name: Case(
            GetLayoutObjectAttribute("tab.mainlist 3";"isFrontTabPanel")  =  1 ; "field.contact.name 3" ;
            GetLayoutObjectAttribute("tab.mainlist 2";"isFrontTabPanel")  =  1 ; "field.contact.name 2" ;
            "field.contact.name"
            ) ]
        Exit Loop If [ 1 ]
    End If
    Commit Records/Requests
        [ Skip data entry validation; No dialog ]
    Exit Loop If [ $$NewItem[4] = "external" ]
    #
    #open
    Perform Script [ “DoubleClicked [ link (optional), $$EnterNote , $$drag.portal]”; Parameter: $link ]
    If [ 1 = 2 ]
        Perform Script [ “ScrollToItemAndSelect [ link, $$drag.portal ]”; Parameter: $link ]
        Set Variable [ $$EnterNote; Value:Case(
            Get(ScriptResult ) = "Failed" ; "key" ;
            Filter (Left( $link ; 1)  ; "AXVYNZ"  ) > 0; "Enter";
            "key"
            ) ]
    End If
    Exit Loop If [ 1 ]
End Loop
#
Loop
    Exit Loop If [ Let( $count = $count + 1 ; $count)   >  12 ]
    Set Field [ window::xSys_NewItem[$count]; "" ]
End Loop
Set Field [ window::xSys_NewItem[12]; $link ]
Set Field [ window::xPt_ItemIDForScript; $lItemp ]
Exit Script [ Result: Let( [
    $$EnterNote = "" ;
    $$drop.note = "" ;
    $$drag.ID = "" ;
    $$drop.ID = "" ;
    $$drag.portal = "";
    $$drag.portalid = "";
    $$drag.note = "";
    $$dragheadername = "";
    $$drop.sidebarID = "" ;
    $$drag.sidebarID = ""  ;
    $$Drag.foldertype = "" ;
    $$Draglist = "" ;
```

```
        $$Dontchange = ""
        ] ; "" ) ]
Show Custom Dialog [ Message: "1 "  & window::xSys_NewItem[ 1 ]  & "¶" &
        "2 "  & window::xSys_NewItem[ 2 ]  & "¶" &
        "3 "  & window::xSys_NewItem[ 3 ]  & "¶" &
        "4 "  & window::xSys_NewItem[ 4 ]  & "¶" &
        "5 "  & window::xSys_NewItem[ 5 ]  & "¶" &
        "6 "  & window::xSys_NewItem[ 6 ]  & "¶¶" &
        "1 $$"  & $$NewItem[1]  & "¶" &
        "2 $$"  & $$NewItem[2] & "¶" &
        "3 $$"  & $$NewItem[3]  & "¶" &
        "4 $$"  & $$NewItem[4]  & "¶" &
        "5 $$"  & $$NewItem[5]  & "¶"&
        "6 $$"  & $$NewItem[6]  & "¶"  &
        "Menu "  & window::xSys_NewMenuChoice  & "¶"; Buttons: “OK”, “Cancel” ]
```