```
on preOpenStack
  set the vScroll of group "PropsGroup" to 0

  switch the platform
  case "MacOS"
    set itemDelimiter to "."
    if item 1 of the systemVersion < 10
    then
      get "203,460"
    else
      if char 1 to 3 of revAppVersion() >= 2.0
      then get "203,460"
      else get "200,462"
    end if
    break

  case "Win32"
    get "202,459"
    break

  default
    if char 1 to 3 of revAppVersion() >= 2.0
    then get "201,460"
    else get "201,458"
    break
  end switch

  set the loc of group "CkeckboxesGroup" to it
end preOpenStack

on resizeStack
  global gFlipper

  if gFlipper is not "true"
  then set the vScroll of group "PropsGroup" to 0
  else delete global gFlipper

  revUpdateGeometry
end resizeStack

--on keyDown what
  global gTypedText,gTypedTime

  if gTypedTime is empty or the ticks > gTypedTime + 30
  then put what into gTypedText
  else put what after gTypedText
  put the ticks into gTypedTime

  repeat
    find gTypedText
    if the foundField is empty then exit repeat

    if the long name of the foundField contains "TitlesGroup"
    then
      lock screen
      set the vScroll of group "PropsGroup" to 0
      repeat until the foundLoc is within the rect of group "PropsGroup"
        set the vScroll of group "PropsGroup" to the vScroll of group "PropsGroup" + 16
      end repeat
      set the vScroll of group "PropsGroup" to the vScroll of group "PropsGroup" + 16
    end if
  end repeat
end keyDown

on closeStack
  CheckForChanges "L"
  CheckForChanges "R"

  lock screen

  repeat with i=1to 230
    put empty into fld i
  end repeat

  UpdateChecks

  set the cObjRef of fld "ObjL" to empty
  set the cObjRef of fld "ObjR" to empty
  set the cChangedNames of fld "ObjL" to empty
  set the cChangedValues of fld "ObjL" to empty
  set the cChangedNames of fld "ObjR" to empty
  set the cChangedValues of fld "ObjR" to empty
  set the cCheckedList of group "CkeckboxesGroup" to empty

  set the enabled of btn "ApplyL" to false
  set the enabled of btn "ApplyR" to false

  set the enabled of group "CkeckboxesGroup" to false
  set the enabled of group "Properties display:" to false
  set the enabled of group "Checkboxes:" to false
end closeStack

on revNameChanged
  UpdateObjet
end revNameChanged

on revIDChanged
  UpdateObjet
end revIDChanged

on revResizeControl
  UpdateObjet
end revResizeControl

on revMoveControl
  UpdateObjet
end revMoveControl

on UpdateObjet
   --  if the long id of the selectedObject is the cObjRef of fld "ObjL"
   --  then GetProps the selectedObject,"L"
   if the long id of the selectedObject is the cObjRef of fld "ObjR"
   then GetProps the selectedObject,"R"
end UpdateObjet

on GetProps theObj,side
  CheckForChanges side
  lock screen

  put the name of theObj into fld ("Obj" & side)
  set the cObjRef of fld ("Obj" & side) to the long id of theObj
  set the cChangedNames of fld ("Obj" & side) to empty
  set the cChangedValues of fld ("Obj" & side) to empty
  set the enabled of btn ("Clear" & side) to true
  set the enabled of btn ("Apply" & side) to false

  DisplayProps

  get (the cObjRef of fld "ObjL" <> empty and the cObjRef of fld "ObjR" <> empty)

  set the enabled of group "CkeckboxesGroup" to it
  set the enabled of group "Properties display:" to it
  set the enabled of group "Checkboxes:" to it
end GetProps
```

```
on ClearProps side
  CheckForChanges side
  lock screen

  put empty into fld ("Obj" & side)
  set the cObjRef of fld ("Obj" & side) to empty
  set the cChangedNames of fld ("Obj" & side) to empty
  set the cCheckedList of group "CkeckboxesGroup" to empty
  set the enabled of btn ("Clear" & side) to false
  set the enabled of btn ("Apply" & side) to false

  repeat with i=1 to number of flds of group (side & "TitlesGroup")
    set the textColor of fld i of group (side & "TitlesGroup") to the backgroundColor of
btn "Matching"
    set the textColor of fld i of group (side & "ValuesGroup") to the backgroundColor of
btn "Matching"
  end repeat

  set the enabled of group "CkeckboxesGroup" to false
  set the enabled of group "Properties display:" to false
  set the enabled of group "Checkboxes:" to false

  DisplayProps
end ClearProps

on DisplayProps
  lock screen
  set the vScroll of group "PropsGroup" to 0

  if the cObjRef of fld "ObjL" is not empty
  then
    put the properties of the cObjRef of fld "ObjL" into LPropsList
    put the keys of LPropsList into LKeysList
    sort lines of LKeysList
  end if

  if the cObjRef of fld "ObjR" is not empty
  then
    put the properties of the cObjRef of fld "ObjR" into RPropsList
    put the keys of RPropsList into RKeysList
    sort lines of RKeysList
  end if

  if LKeysList is not empty and RKeysList is not empty
  then
    set the wholeMatches to true
    repeat with i=max(number of lines of LKeysList,number of lines of RKeysList)
down to 1
      if LineOffset(line i of LKeysList,RKeysList) = 0
      then delete line i of LKeysList
      if LineOffset(line i of RKeysList,LKeysList) = 0
      then delete line i of RKeysList
    end repeat

    put the hilitedButton of group "Properties display:" into displayMode
  else put 1 into displayMode

  put 1 into fldNb

  repeat with i=1 to max(number of lines of LKeysList,number of lines of RKeysList)
    put (LPropsList[line i of LKeysList] = RPropsList[line i of RKeysList]) into
propsMatch

    if (displayMode is 1) or ((displayMode is 2) and propsMatch) or ((displayMode is 3)
and not propsMatch)
    then
      if line i of LKeysList is not empty
      then SetLine fldNb,"L",line i of LKeysList,replaceText(LPropsList[line i of
LKeysList],return,"¶"),the backgroundColor of btn "MatchingColor"
      else SetLine fldNb,"L","","",the backgroundColor of btn "MatchingColor"

      if line i of RKeysList is not empty
      then SetLine fldNb,"R",line i of RKeysList,replaceText(RPropsList[line i of
RKeysList],return,"¶"),the backgroundColor of btn "MatchingColor"
      else SetLine fldNb,"R","","",the backgroundColor of btn "MatchingColor"

      set the visible of btn fldNb of group "CkeckboxesGroup" to (line i of LKeysList is
not in "id,number") and (line i of RKeysList is not in "id,number")

      add 1 to fldNb
    end if
  end repeat

  repeat with i=fldNb to number of flds of group "LTitlesGroup"
    SetLine i,"L","","",the backgroundColor of btn "MatchingColor"
    SetLine i,"R","","",the backgroundColor of btn "MatchingColor"
    hide btn i of group "CkeckboxesGroup"
  end repeat
  if number of lines of the keys of LPropsList = number of lines of LKeysList
  then set the textStyle of fld "ObjL" to "bold"
  else set the textStyle of fld "ObjL" to "bold,italic"

  if number of lines of the keys of RPropsList = number of lines of RKeysList
  then set the textStyle of fld "ObjR" to "bold"
  else set the textStyle of fld "ObjR" to "bold,italic"

  UpdateTextColor
  UpdateChecks
end DisplayProps

on SetLine nb,side,propName,propVal,color
  set wholeMatches to true
  put LineOffset(propName,the cChangedNames of fld ("Obj" & side)) into
changedFound

  put propName into fld nb of group (side & "TitlesGroup")

  if changedFound <> 0
  then
    put word changedFound of the cChangedValues of fld ("Obj" & side) into fld nb of
group (side & "ValuesGroup")
    SetLineColor nb,side
    --    set the textColor of fld nb of group (side & "TitlesGroup") to the
backgroundColor of btn "ChangedColor"
    --    set the textColor of fld nb of group (side & "ValuesGroup") to the
backgroundColor of btn "ChangedColor"
  else
    put propVal into fld nb of group (side & "ValuesGroup")
    SetLineColor nb,side,color
    --    set the textColor of fld nb of group (side & "TitlesGroup") to color
    --    set the textColor of fld nb of group (side & "ValuesGroup") to color
  end if
end SetLine

on UpdateTextColor
  repeat with i=1 to number of flds of group "LTitlesGroup"
    if (fld i of group "LValuesGroup" = fld i of group "RValuesGroup")  \
       or (fld i of group "LValuesGroup" is empty) or (fld i of group "RValuesGroup" is
empty)
    then
      SetLineColor i,"L",the backgroundColor of btn "MatchingColor"
      SetLineColor i,"R",the backgroundColor of btn "MatchingColor"
    else
```

```
      SetLineColor i,"L",the backgroundColor of btn "NonMatchingColor"
      SetLineColor i,"R",the backgroundColor of btn "NonMatchingColor"
    end if
  end repeat
end UpdateTextColor

on SetLineColor nb,side,color
  set wholeMatches to true

  if LineOffset(fld nb of group (side & "TitlesGroup"),the cChangedNames of fld ("Obj" & side)) = 0
  then
    set the textColor of fld nb of group (side & "TitlesGroup") to color
    set the textColor of fld nb of group (side & "ValuesGroup") to color
  else
    set the textColor of fld nb of group (side & "TitlesGroup") to the backgroundColor of btn "ChangedColor"
    set the textColor of fld nb of group (side & "ValuesGroup") to the backgroundColor of btn "ChangedColor"
  end if
end SetLineColor

on UpdateChecks
  set wholeMatches to true
  put false into copyState

  repeat with i=1 to number of btns of group "CkeckboxesGroup"
    get (LineOffset(fld i of group "LTitlesGroup",the cCheckedList of group "CkeckboxesGroup") <> 0)
    put copyState or it into copyState
    set the hilited of btn i of group "CkeckboxesGroup" to it
  end repeat

  set the enabled of btn "CopyL" to copyState
  set the enabled of btn "CopyR" to copyState
end UpdateChecks

on UpdateCopyBtns
  put false into foundChecked
  put empty into checkedList

  repeat with i=1 to number of btns of group "CkeckboxesGroup"
    if the hilited of btn i of group "CkeckboxesGroup"
    then
      put fld i of group "LTitlesGroup" & return after checkedList
      put true into foundChecked
    end if
  end repeat

  set the cCheckedList of group "CkeckboxesGroup" to checkedList

  set the enabled of btn "CopyL" to foundChecked
  set the enabled of btn "CopyR" to foundChecked
end UpdateCopyBtns

on DoApply side
  lock screen

  put the properties of the cObjRef of fld ("Obj" & side) into propsList

  repeat with i=1 to number of flds of group "LTitlesGroup"
    if fld i of group (side & "TitlesGroup") is empty then exit repeat

    put replaceText(fld i of group (side & "ValuesGroup"),"¶",return) into propsList[fld i
of group (side & "TitlesGroup")]
  end repeat

  set the properties of the cObjRef of fld ("Obj" & side) to propsList

  set the enabled of the target to false

  set the cChangedNames of fld ("Obj" & side) to empty
  set the cChangedValues of fld ("Obj" & side) to empty

  UpdateTextColor
end DoApply

on doCopy provSide,destSide
  repeat with i=1 to number of btns of group "CkeckboxesGroup"
    if the hilited of btn i of group "CkeckboxesGroup"
    then
      put (fld i of group (provSide & "ValuesGroup") <> fld i of group (destSide &
"ValuesGroup")) into notMatching

      put fld i of group (provSide & "ValuesGroup") into fld i of group (destSide &
"ValuesGroup")
      set the hilited of btn i of group "CkeckboxesGroup" to false

      if notMatching
      then
```

```
        put the cChangedNames of fld ("Obj" & destSide) into oldChangedNames
        put the cChangedValues of fld ("Obj" & destSide) into oldChangedValues

        put (fld i of group (destSide & "TitlesGroup")) & return after oldChangedNames
        put (fld i of group (destSide & "ValuesGroup")) & return after
oldChangedValues

        set the cChangedNames of fld ("Obj" & destSide) to oldChangedNames
        set the cChangedValues of fld ("Obj" & destSide) to oldChangedValues
      end if
    end if
  end repeat

  set the enabled of btn ("Apply" & destSide) to true

  UpdateTextColor
end doCopy

on ChangeValue side
  put the short name of the target into fldNb

  if (fld fldNb of group (side & "TitlesGroup") is not empty) and (fld fldNb of group
(side & "TitlesGroup") is not in "id,number")
  then
    if target is in "true,false"
    then
      put not target into fld fldNb of group (side & "ValuesGroup")
    else
      switch fld fldNb of group (side & "TitlesGroup")
      case "colors"
      case "patterns"
        get fld fldNb of group (side & "TitlesGroup")
        put toUpper(char 1 of it) into char 1 of it
        set the dialogData to it & return & target
        modal stack "ColorsPatterns"
        if the dialogData is empty then exit ChangeValue
        get replaceText(the dialogData,return,"¶")
        break

      default
        ask "New value of" && fld fldNb of group (side & "TitlesGroup") & ":" with target
        if the result is "Cancel" or it is fld fldNb of group (side & "ValuesGroup") then
exit ChangeValue
        break
      end switch
```

```
   put it into fld fldNb of group (side & "ValuesGroup")
  end if

  do "put the" && fld fldNb of group (side & "TitlesGroup") && "of" && the cObjRef of
fld ("Obj" & side) && "into currentPropVal"
  put the cChangedNames of fld ("Obj" & side) into oldChangedNames
  put the cChangedValues of fld ("Obj" & side) into oldChangedValues
  set wholeMatches to true
  put LineOffset(fld fldNb of group (side & "TitlesGroup"),oldChangedNames) into
changedFound

  if currentPropVal = replaceText(fld fldNb of group (side &
"ValuesGroup"),"¶",return)
  then
   if changedFound <> 0
   then
    delete line changedFound of oldChangedNames
    delete line changedFound of oldChangedValues
   end if

   if fld fldNb of group "LValuesGroup" is fld fldNb of group "RValuesGroup"
   then get "MatchingColor"
   else get "NonMatchingColor"
  else
   if changedFound = 0
   then
    put (fld fldNb of group (side & "TitlesGroup")) & return after oldChangedNames
    put (fld fldNb of group (side & "ValuesGroup")) & return after
oldChangedValues
   end if

   get "ChangedColor"
  end if

  set the cChangedNames of fld ("Obj" & side) to oldChangedNames
  set the cChangedValues of fld ("Obj" & side) to oldChangedValues

  set the textColor of fld fldNb of group (side & "TitlesGroup") to the
backgroundColor of btn it
  set the textColor of fld fldNb of group (side & "ValuesGroup") to the
backgroundColor of btn it

  set the enabled of btn ("Apply" & side) to true
  end if
end ChangeValue

on CheckForChanges side
  if the cChangedNames of fld ("Obj" & side) <> empty
  then
    beep
    answer question "Apply changes to" && fld ("Obj" & side) && "before dismiss?"
with "No" or "Yes"
    if it is "Yes" then DoApply side
  end if
end CheckForChanges
```

```
on mouseUp
  set the htmlText of fld "AboutHelp" to the cAboutHtml of this stack
  show fld "AboutHelp"
  set showBorder of fld "AboutHelp" to false
end mouseUp
```

```
on mouseUp
  DoApply "L"
end mouseUp
```

```
on mouseUp
  DoApply "R"
end mouseUp
```

```
on mouseUp
end mouseUp
```

```
on mouseUp
 lock screen

 repeat with i=1 to number of btns of group "CkeckboxesGroup"
  if the visible of btn i of group "CkeckboxesGroup"
  then
   switch the hilitedButtonName of group "Checkboxes:"
   case "All"
    set the hilited of btn i of group "CkeckboxesGroup" to true
    break

   case "None"
    set the hilited of btn i of group "CkeckboxesGroup" to false
    break

   case "Matching"
    set the hilited of btn i of group "CkeckboxesGroup" to (fld i of group
"LValuesGroup" = fld i of group "RValuesGroup")
    break

   case "Non-matching"
    set the hilited of btn i of group "CkeckboxesGroup" to (fld i of group
"LValuesGroup" <> fld i of group "RValuesGroup")
    break
   end switch
  end if
 end repeat

 UpdateCopyBtns
end mouseUp
```

```
on mouseUp
 ClearProps "L"
end mouseUp
```

```
on mouseUp
 ClearProps "R"
end mouseUp
```

```
on mouseUp
  DoCopy "L","R"
end mouseUp
```

```
on mouseUp
  DoCopy "R","L"
end mouseUp
```

```
on mouseUp
  if the mode of this stack is 1
  then
    if the optionKey is down
    then GetProps the name of this stack,"L"
    else if the shiftKey is down
    then GetProps the name of this cd,"L"
    else GetProps the target,"L"
  else
    if the optionKey is down
    then GetProps the name of the topStack,"L"
    else if the shiftKey is down
    then GetProps the long name of this cd of the topStack,"L"
    else
      if the selectedObject is not empty
      then GetProps the selectedObject,"L"
    end if
  end if
end mouseUp
```

```
on mouseUp
  if the mode of this stack is 1
  then
    if the optionKey is down
    then GetProps the name of this stack,"R"
    else if the shiftKey is down
    then GetProps the name of this cd,"R"
    else  GetProps the target,"R"
  else
    if the optionKey is down
    then GetProps the name of the topStack,"R"
    else if the shiftKey is down
    then GetProps the long name of this cd of the topStack,"R"
    else
      if the selectedObject is not empty
      then GetProps the selectedObject,"R"
    end if
  end if
end mouseUp




on mouseUp
  set the htmlText of fld "AboutHelp" to the cHelpHtml of this stack
  show fld "AboutHelp"
  set showBorder of fld "AboutHelp" to false
end mouseUp




on mouseUp
end mouseUp




on mouseUp
end mouseUp




on mouseUp
  global gFlipper

  put true into gFlipper
  lock screen
```

```
 if the hilite of me
 then
   set the toolTip of me to "Reveal Preferences panel"
   set the cRevGeometry["Master","scaleBottomDistance"] of group "PropsGroup" to
-13
   set the cRevGeometry["Master","movevDistance"] of group "BtnGroup" to -7
   set the cRevGeometry["Master","movevDistance"] of group "Preferences" to 100
   set the rect of this stack to the topLeft of this stack, the right of this stack, the
bottom of this stack - the cDeltaStackHeight of me
   set the minHeight of this stack to 137
 else
   set the toolTip of me to "Hide Preferences panel"
   set the cRevGeometry["Master","scaleBottomDistance"] of group "PropsGroup" to
-13 -the cDeltaStackHeight of me
   set the cRevGeometry["Master","movevDistance"] of group "BtnGroup" to -7 -the
cDeltaStackHeight of me
   set the cRevGeometry["Master","movevDistance"] of group "Preferences" to -81
   set the rect of this stack to the topLeft of this stack, the right of this stack, the
bottom of this stack + the cDeltaStackHeight of me
   set the minHeight of this stack to 137 + the cDeltaStackHeight of me
 end if
end mouseUp
```

```
on mouseUp
 revGoURL "mailto:frederic@runrev.com"
end mouseUp
```

```
on mouseUp
 hide me
end mouseUp
```

```
on mouseUp
 UpdateCopyBtns
end mouseUp
```

```
on mouseUp
  ChangeValue "L"
end mouseUp
```

```
on mouseUp
  DisplayProps
end mouseUp
```

```
on mouseUp
  ChangeValue "R"
end mouseUp
```

```
on mouseUp
  answer color with the backgroundColor of the target
  if the result is not "Cancel"
  then set the backgroundColor of the target to it

  DisplayProps
end mouseUp
```

```
on preopenStack
  set itemDelimiter to "¶"

  set the label of this stack to line 1 of the dialogData

  put the cEntryRoot of fld "Values" into fld "Values"

  repeat with i=1 to 8
    put char 1 to -2 of line 1 of the dialogData after line i of fld "Values"
  if line 1 of the dialogData is "Patterns"
  then
   set the backgroundColor of btn i of group "BtnsGroup" to empty
   set the backgroundPattern of btn i of group "BtnsGroup" to item i of line 2 of the dialogData
  else set the backgroundColor of btn i of group "BtnsGroup" to item i of line 2 of the dialogData
 end repeat
end preopenStack

on openStack
end openStack

on closeStack
end closeStack
```

```
on mouseUp
 set the dialogData to empty
 close this stack
end mouseUp
```

```
on mouseUp
 repeat with i=1 to 8
  if the label of this stack is "Colors"
  then put the backgroundColor of btn i of group "BtnsGroup" into line i of temp
  else put the backgroundPattern of btn i of group "BtnsGroup" into line i of temp
 end repeat

 set the dialogData to temp
 close this stack
end mouseUp
```

```
on mouseUp
 click at loc of btn (word 2 of the clickLine) of group "BtnsGroup"
end mouseUp
```

```
on mouseUp
  if the label of this stack is "Colors"
  then
    answer color with the backgroundColor of the target
    if the result is not "Cancel"
    then
      set the backgroundColor of the target to it
    end if
  else
    set the dialogData to the backgroundPattern of the target
    modal stack "ChoosePattern"
    if the dialogData is not empty
    then
      set the backgroundPattern of the target to the dialogData
    end if
  end if
end mouseUp
```

```
on preOpenStack
  repeat with i=1 to the number of groups of group "PatternsGroup"
    set the filled of group i of group "PatternsGroup" to (i = the dialogData)
  end repeat

  set the vScroll of group "PatternsGroup" to (the dialogData - 1) * 21
end preOpenStack

on openStack
end openStack

on closeStack
end closeStack
```

```
on mouseUp
  set the dialogData to empty
  close this stack
end mouseUp
```

```
on mouseUp
 repeat with i=1 to the number of groups of group "PatternsGroup"
   if the filled of group i of group "PatternsGroup"
   then
     get i
     exit repeat
   end if
 end repeat

 set the dialogData to it
 close this stack
end mouseUp
```

```
on mouseUp
 repeat with i=1 to the number of groups of me
   set the filled of group i of me to (the clickLoc is within rect of group i of me)
 end repeat
end mouseUp

on mouseDoubleUp
 click at loc of btn "Ok"
end mouseDoubleUp
```