

```

;=====
;   EEEEE DDDDD SSSSS
;   EE    DD  DD  SS
;   EEEE  DD  DD  SSSSS
;   EE    DD  DD   SS  (c) 1995 Electronic Data Systems Limited
;   EEEEE DDDDD SSSSS   All Rights Reserved
;=====
;
; File      : <readroot>\READDATA\PROJECTS\DIARY\DYDIARY.VSS
; Author    : Roger C. King
; Date      : 19th September 1995
; Function  : Diary facility diary script task
;
;=====
;
; Project   : Wandsworth
; CCS       : BW1
; Author    : Mayank Ladd
; Date      : March 97
; Version   : Y2000
; Desc      : Call date functions from a script function
;
;=====
;
; Project   : Wandsworth
; CCS       : BW9
; Author    : Mayank Ladd
; Date      : 15 April 1997
; Version   : 1.2
; Desc      : Script \PROJECTS\COMMON\CSDAT00L.VSS not present, loading CSDAT00L.FSL
;
;=====
;
; Project   : Wandsworth
; CCS       : BW12
; Author    : Mayank Ladd
; Date      : 02 May 1997
; Version   : 1.3

```

```

; Desc      : Track Documents within Folders.  Added to track start date for a document
;
;=====
;
; CCS       : CR7                      Author : Simon Burgoyne
; Date      : 10 Oct 97
; Version    : 1.4
; Description : Added forward checks to stop folders ending up in the archive
;              queue with invalid attributes.
;
;=====
;
; CCS       :                      Author : John Cousins
; Date      : 16 May 02
; Version    : 1.5
; Description : Changed default diary exit time to 07:00
;
;=====
; Author     : Tedy Shalev
; Date      : Nov 2004
; Version    : 1.6
; Desc      : Applied new rules when sending HB New Claims worktypes
;              (defined in BRCOM.INI) to Diary.
;              - Inserted condition code in DySetUpDiary
;              - Inserted condition code in DYDiaryPopupOK
;              - Inserted condition code in DYDiaryTask
;              - Added BRNewAppsRules and BRDiaryOKRule
;              - Added to DYSetUpDiary - check if File Note or Word Doc of a
;              list stored in database table are present on folder before
;              allowing the folder to be sent to DIARY.
;              - Added BREnableDiaryCtrl and BRDisableDiaryCtrl
;              - Added BRGetDiaryParam
;              - BRPRIORITY is not reset to "Diary Mature" (= 5) if it is
;              already doneReminder (= 6)
;=====
; Author     : Tedy Shalev
; Date      : 15 Jan 2006
; Version    : 1.6

```

```

; Desc      : No Diarise if folder has no target date.
;=====
; Author    : Tedy Shalev
; Date      : 16 May 2006
; Version   : 1.7
; Desc      : DYDiaryTask - When a folder is sent to diary the target date
;            is set to the diary out date (overrules the calc target date
;            according to the folder's worktype.
;=====
; Author    : Tedy Shalev
; Date      : 24 May 2006
; Version   : 1.8
; Desc      : DYDiaryResetTargetDate
;            When a folder comes out of Diary because it has mail matched
;            to new mail, the folder's target date is set to next working
;            day, as follows:
;            If Today is:
;            Friday    Target Date is Today + 3 days,
;            Saturday  Target Date is Today + 2 days,
;            otherwise Target Date is Today + 1 day only.
;=====
;
; Author    : Tedy Shalev
; Date      : 22 July 2006
; Version   : 1.9
; Desc      : BRGetDiaryParam - Added data for new section in BRCOM.INI
;            Defines a list of worktypes that may be sent to diary for a
;            maximum number of days, limited to same as for the new HB Claims
;            worktypes - this is named: listLimitWorkTypes
;            - DYSetUpDiary - added listLimitWorkTypes
;            - DYDiaryPopupOK - added listLimitWorkTypes
;=====
; Contents:
; -----
; Send to diary functions
; -----
; DYDIARY_Dummy

```

```

; DYDiaryTask
;
; Get from Diary functions
; -----
; DYGetDiary          - Run from the get diary button on Chase.  Displays list of items
;                      in diary.
; DYGetDiaryOpen      - Open the selected document from diary
; DYGetDiarySearch    - Search the diary for documents for the current user
;
;
; Invoked by:
; -----
; Script Task
;
; Scripts/functions called (required preloads):
; -----
; None
;
; Objects loaded/saved:
; -----
; None
;
; Undocumented Viewstar considerations:
; -----
; None
;
;=====

```

```

DefScript DYDIARY_Dummy
MoreArguments ignore

```

```

    DYSetUpDiary nil

```

```

EndScript

```

```

;=====

```

```

; DefScript DYDiaryTask
;   This function is attached to the Action Script on the Diary task.
;   Its job is to set up the workpacket for diarying. It performs the
;   following tasks:
;
;   1. Determines the date/time for return, and places an entry in the
;       Diary table for this request id to be actioned at the specified
;       date/time
; --- TEDY - Nov 2004 ...
;   2. For all new HB claims the original diary in date does not change!
;       All other worktypes - BRDIARYIN date is updated to today's date.
; --- ... TEDY - Nov 2004
; --- TEDY - July 2006 ...
;       Apply same rules as for new HB claims to the list of worktypes
;       listed in BRCOM.INI section: [DiarySpecial] entry: LimitWorkTypes,
;       using listLimitWorkTypes.
; --- ... TEDY - July 2006
;=====
DefScript DYDiaryTask
Arguments abs route
Locals    reqid diarydate dbdate attribs dbm tdreqid tdcurr tdnnext tdwtype tdttdays ~
          tdtdate diarycnt listHBNewApps listLimitWorkTypes

;---BW1 Added---

    Unless (Fboundp 'BRCOMMsg)
        LoadScript (VSRootAppend "readdata\\projects\\br\\script\\brcom.vss") :useful
    EndUnless

;---BW1 Added End---

    Unless (FBoundP 'DYDiarySQLAddEntry)
        LoadScript (VSRootAppend "readdata\\projects\\diary\\dysql.vss")
    EndUnless

    Unless (And (fboundp 'CSPutVal) (fboundp 'CSGetVal))
;       LoadScript (VSRootAppend "PROJECTS\\COMMON\\CSDAT00L.VSS") :mandatory ; BW9 Commented Out
        LoadScript (VSRootAppend "PROJECTS\\COMMON\\CSDAT00L.FSL") :mandatory ; BW9 Added

```

```

EndUnless
;
; Get the diary date and time set on the abstract by the user
; use current values if non assigned

Unless (Assign diarydate (CSGetVal :DIARYDATETIME [[:TARGET abs] [:TOPIC :DIARY]]))
; Assign diarydate (FormatDateTime (GetCurrentDateTime) "dd/MM/yy HH:mm" nil) ;---BW1 Commented Out---
Assign diarydate (BrComDateTimeString (GetCurrentDateTime) "dd/MM/yyyy HH:mm") ;---BW1 Added--- Need to
pass in the format is not HH:mm:ss

EndUnless

; Assign dbdate (ParseDateTime diarydate "dd/MM/yy HH:mm" nil) ;---BW1 Commented Out---
Assign dbdate (BrComInternalDateTime diarydate "dd/MM/yyyy HH:mm") ;---BW1 Added--- Need to pass in the
format is not HH:mm:ss
;
; Create entry in diary database table
;
Assign reqid (Send route :REQ_Q_ID)

DYDiarySQLAddEntry reqid dbdate (GetDocumentInfo abs :DY_DIARYQUE)

;; ===== TEDY - Nov 2004 ... =====
Assign diarycnt (GetDocumentInfo abs :field "BRDIARYCNT")
;; -- Resolve diarycnt type inconsistency...
;; Sometimes string "" and other times numeric 0,1,2,...
If (Type diarycnt '@String)
Then
Assign diarycnt (ReadFromString diarycnt)
EndIf
Unless diarycnt
Assign diarycnt 0
EndUnless

;; ===== TEDY - July 2006 ... =====
;; -- Get list of LimitWorkTypes from env and if not there (1st time) from INI file...
Assign listLimitWorkTypes (BRGetDiaryParam "DiarySpecial" "LimitWorkTypes")
;; ===== ... TEDY - July 2006 =====

```

```

;; -- Get list of New Claims from env and if not there (1st time) from INI file...
Assign listHBNewApps (BRGetDiaryParam "HBReminderLetter" "WorkTypes")
Assign tdwtype (GetDocumentInfo abs :field "BRWKTYPE")

;; ===== TEDY - July 2006 ... =====
;; Include listLimitWorkTypes to same behaviour as the new HB Claims...
; If (Member tdwtype listHBNewApps)
  If (Or (Member tdwtype listHBNewApps) (Member tdwtype listLimitWorkTypes))
;; ===== ... TEDY - July 2006 =====
  Then
    ;; -- Note:
    ;;   The BRDIARYCNT attribute is incremented when the OK button is pressed
    ;;   in the "Send to Diary" dialog, so we check here for "<= 1" rather than
    ;;   for "=0" !!!
    If (<= diarycnt 1)
      Then
;; The following "SetDocumentInfo" remains unchanged...
;; Where there is no BRDIARYIN date, set it to today's date.

;; ===== ... TEDY - Nov 2004 =====

;
; Set the request id and diary date on the document attributes so
; that they are reflected in the diary queue template fields
;
    SetDocumentInfo abs :attribs [{"REQID" reqid]~
                                  ["BRDIARYIN" (GetCurrentDateTime)]~
                                  ["BRDIARYOUT" dbdate]~
                                  ["BRDIARYRSN" (CSGetVal :REASON [[:TARGET abs] [:TOPIC :DIARY]])]}]
;; ===== TEDY - Nov 2004 ... =====
  Else
    ;; -- Already has a BRDIARYIN date, so leave it unchanged!
    SetDocumentInfo abs :attribs [{"REQID" reqid]~
                                  ["BRDIARYOUT" dbdate]~
                                  ["BRDIARYRSN" (CSGetVal :REASON [[:TARGET abs] [:TOPIC :DIARY]])]}]
  EndIf
Else

```

```

        SetDocumentInfo abs :attribs [{"REQID" reqid]~
                                      ["BRDIARYIN" (GetCurrentDateTime)]~
                                      ["BRDIARYOUT" dbdate]~
                                      ["BRDIARYRSN" (CSGetVal :REASON [[:TARGET abs] [:TOPIC :DIARY]])]]]
    EndIf
;; ===== ... TEDY - Nov 2004 =====

;; ===== TEDY - 16 May 2006... =====
;; When a folder is sent to diary set the target date to the diary out date...
    SetDocumentInfo abs :attribs [{"BRTDATE" dbdate]}

;; ===== ... TEDY - 16 May 2006 =====

;
; NB Script tasks MUST return 0 if the workpacket is to be forwarded
; successfully

; *****
; Logging for Target date issues - DATESTRING, TIMESTRING, REQID, CURRENT, NEXT, WTYPE, TDAYS, TDATE
; *****

; Locals tdreqid tdcurr tdnnext tdwtype tdtdays tdtdate

    Assign tdreqid ( GetDocumentInfo abs :field "REQID"      )
    Assign tdcurr  ( GetDocumentInfo abs :CURRENT           )
    Assign tdnnext ( GetDocumentInfo abs :NEXT              )
    Assign tdwtype ( GetDocumentInfo abs :field "BRWKTYPE"  )
    Assign tdtdays ( GetDocumentInfo abs :field "BRTARGET" )
    Assign tdtdate ( BRCComDateTimeString ( GetDocumentInfo abs :field "BRTDATE" ) )

    AppendToFile "K:\\P\\LOG\\TDATE\\TDATE.LOG" ( MakeString (DateString) "," (TimeString) "," tdreqid "," tdcurr
    "," tdnnext "," tdwtype "," tdtdays "," tdtdate "," "DYDiaryTask - End" )

; *****
; End of logging
; *****

Return 0

```

EndScript ;DYDiaryTask

```
;-----
; DYSetUpDiary
; Called from the DIARY button within the application. Pops a
; dialog which the user uses to enter the diary date and time.
;
; ARGUMENTS: ignored
;
; RETURNS:  t or nil
;
; --- TEDY - July 2006 ...
; Added list of worktypes defined in listLimitWorkTypes (BRCOM.INI)
; that would behave similiary to the new HB claims in limiting the
; max number of days that these worktypes may be diarised.
; --- ... TEDY - July 2006
;-----
DefScript DYSetUpDiary
MoreArguments ignore
Locals  (diarydlgfile (VSRootAppend "readdata\\projects\\DIARY\\DIALOGS\\DYDIARY.DIA")) ~
        diarydlg diaryreasons (abs (GetCurrentAbstract :folder t)) (env (GetCurrentEnvir)) ~
        wtypes dtime ddate ret fabswt workType listHBNewApps ~
        attribs tmp1 errmsg vattrs tdreqid tdcurr tdnex tdtwtype tdttdays ~
        tdtdate diaryOUT ctxs docsList docAttribs thisDocType (actioned nil) ~
        listActionDocs daysInDiaryMax listLimitWorkTypes

; -- Load all necessary scripts...
Unless (fboundp 'BRCommMsg)
  LoadScript (VSRootAppend "readdata\\projects\\br\\script\\brcom.vss") :useful
EndUnless

Unless (FBoundP 'DYDiarySQLAddEntry)
  LoadScript (VSRootAppend "readdata\\projects\\diary\\dysql.vss")
EndUnless

Unless (And (fboundp 'CSGetVal) (fboundp 'CSPutVal))
  LoadScript (VSRootAppend "PROJECTS\\COMMON\\CSDAT00L.FSL") :mandatory
```

EndUnless

Unless (FBoundP 'BRSQLExecSP)

LoadScript (VSRootAppend "readdata\\projects\\br\\scripts\\brsql.vss")

EndUnless

; -- Check for a folder..

Unless abs

BRCOMMsg "You do not have a folder open to send to the diary" "DYDIARY.VSS" "DYSetUpDiary"

Return nil

EndUnless

;; ===== TEDY - July 2006 ... =====

;; -- Get list of LimitWorkTypes from env and if not there (1st time) from INI file...

Assign listLimitWorkTypes (BRGetDiaryParam "DiarySpecial" "LimitWorkTypes")

;; ===== ... TEDY - July 2006 =====

;; ===== TEDY - Nov 2004 ... =====

;; -- For all New Claims types ONLY (refer to setting in BRCOM.INI file) !

;; Would allow sending the currently opened folder to diary IF the folder

;; contained at least one of the Word documents that are added when a new claim

;; is actioned (refer to new database table HBClaimsWordDocs), or IF the folder

;; has had a File Note attached (irrelevant of its contents!)

;; -- Get list of New Claims from env and if not there (1st time) from INI file...

Assign listHBNewApps (BRGetDiaryParam "HBReminderLetter" "WorkTypes")

Assign workType (GetDocumentInfo abs :field "BRWKTYPE")

When (Member workType listHBNewApps)

;; -- Get list of valid action docs from table HBClaimsWordDocs (in PRES DAT)

Assign listActionDocs (BRSQLExecSP "sp_brHBClaimsWordDocs")

;; -- Get list of documents in folder from the Table of Contents...

Assign docsList (GetFolderTOC (GetCurrentViewDoc :folder t))

DoList (thisDoc docsList)

When (Or (Member thisDoc listActionDocs) ~

(StringSearch "File Note" (First thisdoc)))

;; -- Found one of our "action" docs, no need to look further, so

```
;; break out of this DoList loop...
Assign actioned t
Return
EndWhen
EndDoList
```

```
Unless actioned
;BRCOMMsg "Please refer or action this folder first!" "DYDIARY.VSS" "DYSetUpDiary"
BRCOMShowMessageBox :title "Send To Diary" ~
:text ["Unable to send to Diary." "" ~
"Either refer or action this folder first !" ""] ~
:BUTTONS :OK :ICON :EXCLAIM
```

```
SYS::BEEP
Return
EndUnless
EndWhen
```

```
;; ===== ... TEDY - Nov 2004 =====
```

```
; Validate against lib template. We do this here rather than the fwd save so that the user does not ;SB1.4
Added
; have to keep configuring the diary dialog. ;SB1.4
Added
```

```
Assign attrs (GetAbstractAttrs abs) ;SB1.4
Added
Assign tmpl (GetIndexTemplate (or (send abs :ABS_Q_INDEX) @*TOP*)) ;SB1.4
Added
Assign errmsg (CheckAttrs attrs tmpl) ;SB1.4
Added
```

```
When errmsg ;SB1.4
Added
Return errmsg ;SB1.4
Added
EndWhen ;SB1.4
Added
```

```
; Get diary dialog
```

```

Unless (Assign diarydlg (CSGetVal :DIARY [[:TARGET env] [:TOPIC :BRDIALOGS]] ))

    Unless (Assign diarydlg (ObjLoad diarydlgfile))

        BRCommMsg "Failed to locate Diary Dialog." "DYDIARY.VSS" "DYSetUpDiary" 2
        Return nil

    EndUnless

    CSPutVal :DIARY diarydlg [[:TARGET env] [:TOPIC :BRDIALOGS]]

EndUnless

; Get\populate diary reasons combo box.
; Unless (Assign diaryreasons (CSGetVal :BRDIARYREASONS [[:TARGET env] [:TOPIC :CONFIG_VALS]]))

    Unless (Fboundp 'BRSQLExecSP)
        LoadScript (VSRootAppend "readdata\\projects\\br\\script\\brsql.vss") :infreq
    EndUnless

    Assign diaryreasons (BRSQLExecSP "sp_brseldreason")

    Send (Send diarydlg :DIALOG_Q_CTRL "defres") :CBOX_S_BUFFER diaryreasons

    CSPutVal :BRDIARYREASONS diaryreasons [[:TARGET env] [:TOPIC :CONFIG_VALS]]

; EndUnless

; Populate mature to combo box

Send (Send diarydlg :DIALOG_Q_CTRL "actions") :CBOX_S_BUFFER [["Archive"] ["Workflow"]]

; Reset Controls
DialogSetItem diarydlg "res" "Awaiting Further Information"
DialogSetItem diarydlg "retdate" ""
DialogSetItem diarydlg "time" ""
Send (send diarydlg :DIALOG_Q_CTRL "defres") :CBOX_S_PICK nil
DialogSetItem diarydlg "mag" ""

```

[illegible]

```

Unless (Assign wtypes (CsGetVal :BRWORKTYPES [[:TARGET env] [:TOPIC :CONFIG_VALS]]))
  Unless (Fboundp 'BRCOMSetupAppCustomData)
    LoadScript (VSRootAppend "readdata\\projects\\br\\script\\brcom.vss") :useful
  EndUnless
  BRCOMSetupAppCustomData
  Assign wtypes (CsGetVal :BRWORKTYPES [[:TARGET env] [:TOPIC :CONFIG_VALS]])
EndUnless

; Set default (no folder worktype) to 28 days
; Assign dtime (FormatDateTime (GetCurrentDateTime) "HH:mm" nil)
; Assign ddate (FormatDateTime (DateTime+ (GetCurrentDateTime) 28 :DAY) "dd/MM/yy" nil) ;---BW1 Commented
Out---
  Assign ddate (BrComDateString (DateTime+ (GetCurrentDateTime) 28 :DAY)) ;---BW1 Added---
  Assign dtime "07:00" ;---JC Added---

When (Assign fabswt (GetDocumentInfo (GetCurrentAbstract :folder t) :field "BRWKTYPE"))
  Dolist (tmp wtypes)
    When (StringEqual fabswt (Second tmp))
      Assign ret (DYCalcDateTime (Nth 7 tmp))
      ;Assign dtime (First ret) ;Get Time
      Assign dtime "07:00" ;---JC Added---
      Assign ddate (Second ret) ;Get Date
      If (StringEqual (Nth 8 tmp) "A") ;Set mature to archive on.
        Then
          Send (Send diarydlg :DIALOG_Q_CTRL "actions") :CBOX_S_PICK (@position ["Archive"] ( Send (Send
diarydlg :DIALOG_Q_CTRL "actions") :CBOX_Q_BUFFER) :test '@equal)
          Send (Send diarydlg :DIALOG_Q_CTRL "cbox2") :WND_S_ENABLED :NO
        Else
          Send (Send diarydlg :DIALOG_Q_CTRL "actions") :CBOX_S_PICK (@position ["Workflow"] ( Send (Send
diarydlg :DIALOG_Q_CTRL "actions") :CBOX_Q_BUFFER) :test '@equal)
          Send (Send diarydlg :DIALOG_Q_CTRL "cbox2") :WND_S_ENABLED :YES
        Endif
      When (StringEqual (Nth 10 tmp) "N")
        Send (Send diarydlg :DIALOG_Q_CTRL "actions") :WND_S_ENABLED :NO ;Stop users changing mature
destn.
      EndWhen
    Return
  EndWhen

```

```

        EndDoList
    EndWhen

EndIf

DialogSetItem diarydlg "retdate" ddate
DialogSetItem diarydlg "time" dtime

; Display the diary setup interface

Unless (BRCOMPopup diarydlg)
    Return nil
EndUnless

; Forward the document to the diary

ShowMessage ["Forwarding \. \. \."]

Unless (fboundp 'BRCOMForward')
    LoadScript (VSRootAppend "readdata\\projects\\br\\script\\brcom.vss") :useful
EndUnless

; *****
; Logging for Target date issues - DATESTRING, TIMESTRING, REQID, CURRENT, NEXT, WTYPE, TDAYS, TDATE
; *****

; Locals tdreqid tdcurr tdnnext tdwtype tdtdays tdtdate

Assign tdreqid ( GetDocumentInfo abs :field "REQID" )
Assign tdcurr :WORKBENCH
Assign tdnnext :DIARY
Assign tdwtype ( GetDocumentInfo abs :field "BRWKTYPE" )
Assign tdtdays ( GetDocumentInfo abs :field "BRTARGET" )
Assign tdtdate ( BRCOMDateTimeString ( GetDocumentInfo abs :field "BRTDATE" ) )

AppendToFile "K:\\P\\LOG\\TDATE\\TDATE.LOG" ( MakeString (DateString) "," (TimeString) "," tdreqid "," tdcurr
"," tdnnext "," tdwtype "," tdtdays "," tdtdate "," "DYSetUpDiary - End" )

```

```

; *****
; End of logging
; *****

;; -----
;; No Diarise if folder has no target date.
;; Tedy - 15 Jan 2006
;; -----
Unless (GetDocumentInfo abs :field "BRTDATE")
  SYS::BEEP
  BRComMsg "Folder has no target date! Please call System Administrator." "BRREFER.VSS" "BRRefer" 2
  Return nil
EndUnless
;; -----

BRComForward :WORKBENCH :DIARY

Return t

EndScript ;DYSetUpDiary

;-----
;
; DYDiaryMatureCBox
;
; Called from the FDATA hook on the Mature to combo-box on the Diary
; Setup dialog.
;
; ARGUMENTS: cboc dlg ignore
; RETURNS:
;-----
DefScript DYDiaryMatureCBox

  Arguments cboc dlg ignore

  locals row

; Get cboc row selected.

```

```
Unless (Assign row (send cbox :CBOX_Q_ROW (send cbox :CBOX_Q_PICK)))
  Return nil
EndUnless
```

```
If (Equal (First row) "Archive")
Then
  DialogSetItem dlg "cbox2" nil
  Send (Send dlg :DIALOG_Q_CTRL "cbox2") :WND_S_ENABLED :NO
Else
  Send (Send dlg :DIALOG_Q_CTRL "cbox2") :WND_S_ENABLED :YES
Endif
```

```
Return t
```

```
EndScript
```

```
;-----
;
; DYDiaryDefaults:
; Called from the FDATA hook on the Default Reasons combo-box on the Diary
; Setup dialog. Sets the Return Date and Time fields on the dialog as
; associated with the diary reasons.
;
; ARGUMENTS: cboc dlg ignore
; RETURNS:
;-----
```

```
DefScript DYDiaryDefaults
```

```
Arguments cbox dlg ignore
```

```
Locals durn ddate dtime ret
```

```
;
; Get the diary duration associated with the default
; diary reason chosen
;
```

```
Unless (fboundp 'BRCOMMsg)
```

```

    LoadScript (VSRootAppend "readdata\\projects\\br\\script\\brcom.vss") :useful
EndUnless

Unless (Assign durn (Second (send cbox :CBOX_Q_ROW (send cbox :CBOX_Q_PICK))))
    BRCOMShowMessageBox :title "Diary Setup" :text ["Cannot get default settings for diary reason selected."]
:ICON :EXCLAIM :BUTTONS :OK
    Return nil
EndUnless

If (StringEqual durn "FM") Then
;
; Special case of diary until first of the next month
;
    Assign ddate (DYFirstofMonth)
    Assign dtime (FormatDateTime (GetCurrentDateTime) "HH:mm" nil)

Else

    Assign ret (DYCalcDateTime durn)
    ;Assign dtime (First ret)
    Assign dtime "07:00" ;---JC Added---
    Assign ddate (Second ret)

EndIf

DialogSetItem dlg "res" (First (send cbox :CBOX_Q_ROW (send cbox :CBOX_Q_PICK)))
DialogSetItem dlg "retdate" ddate
DialogSetItem dlg "time" dtime

EndScript

;-----
;
; DYCalcDateTime:
; Calculates a date from the increment string passed e.g "Y99" today+ 99 years.
;
; ARGUMENTS: durn          increment.

```

```

; RETURNS:  dtime          destination time.
;           ddate          destination date.
;-----
DefScript DYCalcDateTime

    Arguments durn

    Locals dtime ddate dypriority

;---BW1 Added---

    Unless (Fboundp 'BrComDateString)
        LoadScript (VSRootAppend "readdata\\projects\\br\\script\\brcom.vss") :useful
    EndUnless

;---BW1 Added End---

;
; Use the actual time increment from the standard reason to calc the destination date and time.
;
; Assign dtime (FormatDateTime (GetCurrentDateTime) "HH:mm" nil)
;   Assign dtime "07:00" ;---JC Added---
; Assign ddate (FormatDateTime (GetCurrentDateTime) "dd/MM/yy" nil) ;---BW1 Commented Out---
;   Assign ddate (BrComDateString (GetCurrentDateTime)) ;---BW1 Added---

    Cond
        Case (Equal (SubString durn 0 1) "Y")
;           Assign ddate (FormatDateTime (DateTime+ (GetCurrentDateTime) (ReadFromString(Substring durn 1)) :YR)
"dd/MM/yy" nil) ;---BW1 Commented Out---
;           Assign ddate (BrComDateString (DateTime+ (GetCurrentDateTime) (ReadFromString(Substring durn 1))
:YR)) ;---BW1 Added---
            EndCase
        Case (Equal (SubString durn 0 1) "N")
;           Assign ddate (FormatDateTime (DateTime+ (GetCurrentDateTime) (ReadFromString(Substring durn 1)) :MON)
"dd/MM/yy" nil) ;---BW1 Commented Out---
;           Assign ddate (BrComDateString (DateTime+ (GetCurrentDateTime) (ReadFromString(Substring durn 1))
:MON)) ;---BW1 Added---
            EndCase

```

```

        Case (Equal (SubString durn 0 1) "D")
;      Assign ddate (FormatDateTime (DateTime+ (GetCurrentDateTime) (ReadFromString(Substring durn 1)) :DAY)
"dd/MM/yy" nil) ;---BW1 Commented Out---
        Assign ddate (BrComDateString (DateTime+ (GetCurrentDateTime) (ReadFromString(Substring durn 1))
:DAY)) ;---BW1 Added---
        EndCase
        Case (Equal (SubString durn 0 1) "H")
        Assign dtime (FormatDateTime (DateTime+ (GetCurrentDateTime) (ReadFromString(Substring durn 1)) :HR)
"HH:mm" nil)
        EndCase
        Case (Equal (SubString durn 0 1) "M")
        Assign dtime (FormatDateTime (DateTime+ (GetCurrentDateTime) (ReadFromString(Substring durn 1)) :MIN)
"HH:mm" nil)
        EndCase
        Otherwise
        ;Do Nothing
        EndCase
    EndCond

    Return [dtime ddate]

```

EndScript

```

;-----
;
; DYDiaryPopupOK
; Called from the OK button on the Diary Setup dialog. Modifies the abstract
; attribs and sets the :DIARY :DIARYDATE, :DIARYTIME properties on the abstract.
;
; ARGUMENTS: ignore dlg ignore
; RETURNS:
;
; --- TEDY - July 2006 ...
; Added list of worktypes defined in listLimitWorkTypes (BRCOM.INI)
; that would behave similary to the new HB claims in limiting the
; max number of days that these worktypes may be diarised.
; --- ... TEDY - July 2006
;-----

```

```

DefScript DYDiaryPopupOK
Arguments ignore dlg ignore
Locals      (abs (GetCurrentAbstract :folder t)) ~
             (retdate (StringTrim (DialogGetItem dlg "retdate") :both)) ret ~
             (rettime (StringTrim (DialogGetItem dlg "time") :both)) ~
             (res (StringTrim (DialogGetItem dlg "res") :both)) (dcnt 0) ~
             rcboxrow (env (GetCurrentEnvir)) msg tdate interval tdate2 ~
             interval2 maturecbox dypriority brwktype testdate ~
             workType listHBNewApps (doneReminder 6) prio listLimitWorkTypes

;; =====
;; TEDY - Nov 2004 ...
;; Apply HB New Application worktypes rules...
;; Only applies to worktypes defined in BRCOM.INI
;; Apply ceiling to return from Diary date [daysInDiaryMax days] - in BRCOM.INI
;; No date extension for diary unless by Team Leaders
;; ... TEDY - Nov 2004
;; =====
Assign workType (GetDocumentInfo abs :field "BRWKTYPE")

;; -- Get list of New Claims from env and if not there (1st time) from INI file...
Assign listHBNewApps (BRGetDiaryParam "HBReminderLetter" "WorkTypes")

;; ===== TEDY - July 2006 ... =====
;; -- Get list of LimitWorkTypes from env and if not there (1st time) from INI file...
Assign listLimitWorkTypes (BRGetDiaryParam "DiarySpecial" "LimitWorkTypes")

; If (Member workType listHBNewApps)
;   If (Or (Member workType listHBNewApps) (Member workType listLimitWorkTypes))
;; ===== ... TEDY - July 2006 =====
;   Then
;     Unless (BRDiaryOKRule abs dlg retdate rettime)
;       Return
;     EndUnless
;   EndIf

;; =====

```

```

; Unless (fboundp 'BRCOMMMsg) ;AB1.1 - Commented Out
  Unless (And (fboundp 'BRCOMMMsg) (fboundp 'BRCOMTargetDate) ) ;AB1.1 - Added
    LoadScript (VSRootAppend "readdata\\projects\\br\\script\\brcom.vss") :useful
  EndUnless

  Unless (And (fboundp 'CSPutVal) (fboundp 'CSGetVal))
;    LoadScript (VSRootAppend "PROJECTS\\COMMON\\CSDAT00L.VSS") :useful ; BW9 Commented Out
    LoadScript (VSRootAppend "PROJECTS\\COMMON\\CSDAT00L.FSL") :useful ; BW9 Added
  EndUnless

; No dairy reason

  Unless res
    BRCOMShowMessageBox :title "Diary Setup" :text ["You must enter a diary reason."] :ICON :EXCLAIM :BUTTONS
:OK
  EndUnless

; No date specified

; Unless (ParseDateTime retdate "dd/MM/yy" nil) ;---BW1 Commented Out---
  Unless (BrComInternalDate retdate) ;---BW1 Added---
    BRCOMShowMessageBox :title "Diary Setup" :text ["Specify A Valid Diary Date"] :ICON :EXCLAIM :BUTTONS :OK
    Return nil
  EndUnless

; No time, use current

  Unless (ParseDateTime rettime "HH:mm" nil)
    Assign rettime (FormatDateTime (GetCurrentDateTime) "HH:mm" nil)
  EndUnless

; These values will be used by the first task in the diary route
; which sets up the diary task (the second in the diary route)

CSPutVal :DIARYDATETIME (MakeString retdate " " rettime) [[:TARGET abs] [:TOPIC :DIARY]]
CSPutVal :REASON res [[:TARGET abs] [:TOPIC :DIARY]]

; Remove any previous expiry flags

```

```
SetDocumentInfo abs :DY_DIARYEXPIRED nil
```

```
; Process mature to combo box.
```

```
When (Assign maturecbx (Send dlg :DIALOG_Q_CTRL "actions"))
```

```
    Cond
```

```
        Case (StringEqual (First (Send maturecbx :CBOX_Q_ROW)) "Archive")
            CSPutVal :MATURETO :ARCHIVE [[:TARGET abs] [:TOPIC :DIARY]]
        EndCase
```

```
        Otherwise
            CSPutVal :MATURETO :WORKFLOW [[:TARGET abs] [:TOPIC :DIARY]]
        EndCase
```

```
    EndCond
```

```
EndWhen
```

```
; Process return to me check box
```

```
When (Equal (DialogGetItem dlg "cbx2") 0)
    SetDocumentInfo abs :BRUSERID (First (Whoami))
EndWhen
```

```
; Increment diary count
```

```
If (Assign dcnt (GetDocumentInfo abs :field "BRDIARYCNT"))
Then
    If (Type dcnt '@Integer')
    Then
        SetDocumentInfo abs :attribs [[:"BRDIARYCNT" (+ dcnt 1)]]
    Else
        SetDocumentInfo abs :attribs [[:"BRDIARYCNT" 1]]
    Endif
Else
```

```
SetDocumentInfo abs :attribs [{"BRDIARYCNT" 1}]
EndIf
```

; Adjust the folder target date from the folder worktype. If we cannot get one make a default one up and see if its accetable.

```
When (Type (Assign tdate (GetDocumentInfo abs :field "BRTDATE")) '@String)
; Assign tdate (ParseDateTime tdate "dd/MM/yy" nil) ;---BW1 Commented Out---
; Assign tdate (BrComInternalDate tdate) ;---BW1 Added---
EndWhen
```

; AB1.1 New Code Added

```
; if the diary return date is equal todays date the target date is not changed
; If the folder target date is less than the diary date
; then the folder target date is set as follows
; target date = retdate + the target interval associated with the folders worktype
```

```
Assign testdate (BrComInternalDate (GetCurrentDateTime))
```

```
When ( DateTime> (BrComInternalDate retdate) testdate :day )
; When ( DateTime< tdate (BrComInternalDate retdate) :day ) ;---BW1 Added---
```

```
Assign brwktype (GetDocumentInfo abs :field "BRWKTYPE")
```

```
Unless ( Assign interval (Nth 5 ( Lookup ( CSGetVal :BRWORKTYPES2 [[:TARGET env] [:TOPIC :CONFIG_VALS]] )
brwktype ) ) )
; BRCOMShowMessageBox :title "Diary Setup" :text ["Cannot find target interval for the Worktype on the
current folder." "Ensure the folder has a valid Worktype."] :ICON :EXCLAIM :BUTTONS :OK
Return nil
EndUnless
```

```
; SetDocumentInfo abs :attribs [{"BRTDATE" (BRCOMTargetDate (ParseDateTime retdate "dd/MM/yy" nil)
interval)]] ;---BW1 Commented Out---
; SetDocumentInfo abs :attribs [{"BRTDATE" (BRCOMTargetDate (BrComInternalDate retdate) interval)]]
;---BW1 Added--
```

```

EndWhen
EndWhen
; AB1.1 End of Added Code

```

```

#| AB1.1 Commented Out

```

```

Assign interval (GetDocumentInfo abs :field "BRDYTARGET")

```

```

If (And tdate interval)
Then

```

```

    SetDocumentInfo abs :attribs [["BRTDATE" (DateTime+ tdate interval :Day)]]

```

```

Else

```

```

    Unless tdate
        Assign tdate2 (GetCurrentDateTime)
    EndUnless

```

```

    Unless Interval
        Assign Interval2 28
    EndUnless

```

```

    SYS::BEEP

```

```

    Unless (BRCOMShowMessageBox :title "Diary Setup Warning" ~
                                :text [(MakeString " Could not update folder target completion
date.") ~
                                (MakeString " Target completion date is " tdate "
Interval " Interval " ".") ~
                                ""~
                                (MakeString " Do you wish to use the default values?" )
~
                                (MakeString " Target date " (Or tdate tdate2) ",
Interval 28 days.")~
                                :buttons :OKCAN ~
                                :icon :QUEST)

```

```

;      Assign tdate tdate2
      SetDocumentInfo abs :attribs [["BRTDATE" (DateTime+ (Or tdate tdate2) interval :Day)]]
      EndUnless

      EndIf

|# ;AB1.1 End of Commented Out section

; Set the priority of diary items

Unless (Assign dypriority (CSGetVal :DIARYPRIORITY [[:TARGET (GetCurrentEnvir)] [:TOPIC :DIARY]]) )
  Assign dypriority (First (First (BRSQLExecSP "sp_brselpriority2 DiaryMature" nil) ) )
  CSPutVal :DIARYPRIORITY dypriority [[:TARGET (GetCurrentEnvir)] [:TOPIC :DIARY]]
EndUnless

;; ===== TEDY - Nov 2004 ... =====
;; -- Condition the setting of Priority = "DiaryMature" (5) with
;; the fact that the folder has not had a reminder letter added.
Assign prio (GetDocumentInfo abs :field "BRPRIORITY")
If (Type prio '@String)
  Then
    Assign prio (ReadFromString prio)
  EndIf
Unless (= prio doneReminder)
  SetDocumentInfo abs :attribs [["BRPRIORITY" dypriority]]
EndUnless
;; ===== ... TEDY - Nov 2004 =====

; Decide which diary queue to send the document to and set property

SetDocumentInfo abs :DY_DIARYQUE "BRDIARY"

ReturnFromPopup t

EndScript ;DYDiaryPopupOK

;=====

```

DefScript DYDiaryPopupCBoxSetDate

; Function attached to diary date/time selection checkboxes. It increments
; the diary date/time based on the selection and increment added.

Arguments ignore dlg ignore

Locals mag pick diarydate date time

;---BW1 Added---

Unless (Fboundp 'BrComDateString)
 LoadScript (VSRootAppend "readdata\\projects\\br\\script\\brcom.vss") :useful
EndUnless

;---BW1 Added End---

; Get the selected checkbox setting

Assign pick (DialogGetItem dlg "cbox")

; If a checkbox was selected check the number of units is set and valid

When pick

; Get the number of units to add to set the date/time

Assign mag (ReadFromString (DialogGetItem dlg "mag"))
Unless (And (Type mag '@number') (> mag 0))
 Assign mag 0
 DialogSetItem dlg "mag" 0
EndUnless
EndWhen

; Depending on what checkbox was selected (if any), set the date/time

```

Switch pick
  Case 0 ;Hours selected
    Assign diarydate (DateTime+ (GetCurrentDateTime) mag :HR)
  EndCase
  Case 1 ;Days selected
    Assign diarydate (DateTime+ (GetCurrentDateTime) mag :DAY)
  EndCase
  Case 2 ;Weeks selected
    Assign diarydate (DateTime+ (GetCurrentDateTime) (* mag 7) :DAY)
  EndCase
  Case 3 ;Months selected
    Assign diarydate (DateTime+ (GetCurrentDateTime) mag :MON)
  EndCase
  Otherwise ;No box selected
    Assign diarydate (GetCurrentDateTime)
  EndCase
EndSwitch

```

```

; Populate the dialogue fields with the new settings

```

```

Switch pick
  Case 0 ;Hours selected
    Assign date (BrComDateString diarydate)
    Assign time (FormatDateTime diarydate "HH:mm" nil)
  EndCase
  Otherwise
    Assign date (BrComDateString diarydate)
    Assign time "07:00"
  EndCase
EndSwitch

```

```

; Assign date (FormatDateTime diarydate "dd/MM/yy" nil)    ;---BW1 Commented Out---
;   Assign date (BrComDateString diarydate)    ;---BW1 Added---
; Assign time (FormatDateTime diarydate "HH:mm" nil)

```

```

DialogSetItem dlg "retdate" date
DialogSetItem dlg "time" time

```

Return t

EndScript

```
;-----
;
; DYFirstOfMonth
;   This script determines the first day of the next month.
;
; ARGUMENTS:
;   date - a string in dd/mm/yy format (optional)
;
; RETURNS: the date of the first of the next month as a string
;-----
DefScript DYFirstOfMonth
Optionals (date (DateString))
Locals newdate

;---BW1 Added---

Unless (Fboundp 'BrComDateString)
  LoadScript (VSRootAppend "readdata\\projects\\br\\script\\brcom.vss") :useful
EndUnless

;---BW1 Added End---

;- Format using VS internal format.
; Unless (Assign date (ParseDateTime date "dd/MM/yy" nil)) ;---BW1 Commented Out---
;   Unless (Assign date (BrComInternalDate date)) ;---BW1 Added---
;   Return nil
;   EndUnless

;- Add a month.
Assign newdate (DateTime+ date 1 :MON)
; Assign newdate (FormatDateTime newdate "dd/MM/yy" nil) ;---BW1 Commented Out---
;   Assign newdate (BrComDateString newdate "dd/MM/yyyy" nil) ;---BW1 Added---
;Inspect ["parsed formatted" date newdate]
```

```

;- Change the day to the first.
Assign newdate (MakeString "01" (SubString newdate 2))

Return newdate
EndScript ;DYFirstOfMonth

;*****
;
; DYGetDiary:
; Called from GetDiary button. Allows the user to search and review the dairy
; queue
;
; ARGUMENTS: none
; RETURNS: t/nil
;*****
DefScript DYGetDiary

Use "Access"

MoreArguments ignore

Locals (dlgfile (VSRootAppend "READDATA\\PROJECTS\\DIARY\\DIALOGS\\DYGETDRY.CP")) ~
      dlg opm tmp1 datalst abs diariedfrom opmgr freefields dlsearch (env (GetCurrentEnvir))~
      team

Unless (FBoundP 'DYDiarySelectExpired)
  LoadScript (VSRootAppend "readdata\\projects\\diary\\dysql.vss")
EndUnless

When (GetCurrentAbstract)
  ShowMessage ["You cannot open another folder until you process the current folder"]
  SYS::BEEP
  Return
EndWhen

Unless (And (fboundp 'CSGetVal) (fboundp 'CSPutVal))

```

```

    LoadScript (VSRootAppend "projects\\common\\csdatool.f4s")
EndUnless

; Get dialog

If (Assign dlg (CSGetVal :DYGETDRY [[:TARGET env] [:TOPIC :BRDIALOGS]]))
Then

    Assign datalist (send dlg :DIALOG_Q_DL "DYGETDIARY")

EndIf
; Else

Unless (Assign dlg (ObjLoad dlgfile))
    BRCommMsg "Failed to locate get from diary Dialog." "DYGETDRY.VSS" "DYGetDiarY" 2
    Return nil
EndUnless

; Adjust the dlg object to have edit fields and to display
; the returned buffer in a different order

Assign datalist (send dlg :DIALOG_Q_DL "DYGETDIARY")
Assign tmpl (send '@namemgr_create :NAME_GET "TEMPLATE" "BRDIARY")

UpdateDatalist datalist tmpl dlg nil

;   Send datalist :DL_S_DISPLAY_LIST [8 9 0 1 2 3 16 15 5 6 18 23 24]
    Send datalist :DL_S_DISPLAY_LIST [0 1 3 22 23 24 6 5 8 9 31]

; Reset dialog

Send datalist :DL_S_BUFFER nil
Send datalist :DL_S_STRING nil

CSPutVal :DYGETDRY dlg [[:TARGET env] [:TOPIC :BRDIALOGS]]

; Endif

```

```

; Reset dialog message box.

DialogSetItem dlg "diarymsg" ""

; Set the search criteria on the datalist

Assign team (SubString (First (Send (Send (GetActionBar "Main") :DIALOG_Q_CTRL "BRINBOX") :CBOX_Q_ROW)) 2)

When (Assign dlsearch (Send dlg :OBJ_GETPROP :SEARCH_CRITERIA))
  DoList (crit dlsearch)
    Send datalist :DL_S_STRING (First crit) (Second crit)
  EndDoList
EndWhen

Send datalist :DL_S_STRING "BRTEAM" team

; Search for entries.

DYGetDiarySEARCH nil dlg nil

; Popup up the dialog.

Unless (BRCOMPopup dlg)

Return nil
EndUnless

Assign datalist (send dlg :DIALOG_Q_DL "DYGETDIARY")

ShowBusyCursor

; Get the open manager we will use to access the diary

Unless (Assign opm (CSGetVal :BRQDIARY [[:TOPIC :OPM]]))
  Assign opm ( CreateDevice "QUEUE" )
  Send opm :OP_GOTO "BRDIARY"
  CSPutVal :BRQDIARY opm [[:TOPIC :OPM]]
EndUnless

```

```

; Note: using a saved property on the datalist object rather than the buffer as the buffer is
; modified for display purposes
Unless (Assign abs (First (send opm :OP_OPEN (send datalist :OBJ_GETPROP :DLBUFF )(send datalist :DL_Q_PICK)
'@EDIT)))
  ShowMessage["The selected document could not be opened"]
  SYS::BEEP
EndUnless

ShowMessage ["Opening \. \. \."]

; Open the folder into the current application.

Send (GetCurrentEnvir) :FWDENV_OPEN abs opm

EndScript

;*****
;
; DYGetDiarySEARCH:
; Called from SEARCH button on the Get From Diary dialog.
; Searches the diary queue for current user and worktype. Uses any additional
; in the datalist edit fields as search attributes.
;
; ARGUMENTS: ignore dlg ignore
; RETURNS:  n/a
;*****
DefScript DYGetDiarySEARCH

Arguments ignore dlg ignore

Use "Access" "SECURITY"

Locals opm tmp1 searchattrs msg1 msg2 buffer (buff2 []) dl team~
(env (GetCurrentEnvir))

DialogSetItem dlg "diarymsg" ""

```

```

; Get the open manager we will use to access the diary

Unless (Assign opm (CSGetVal :BRQDIARY [[:TOPIC :OPM]]))
  Assign opm ( CreateDevice "QUEUE" )
  Send opm :OP_GOTO "BRDIARY"
  CSPutVal :BRQDIARY opm [[:TOPIC :OPM]]
EndUnless

Assign tmp1 (send opm :OP_Q_TMPL)

; If the user has entered any details into the datalist edit fields, then we need
; to format the search criteria based on them. We also need to add team into the search
; Criteria as this is mandatory.

Assign dl (GetDataList dlg)
Assign searchattrs (Append [["STATUS" 0 ]] (GetEditFields dl) )
Assign msg1 "Searching diary..."
Assign msg2 (MakeString "No folders found in diary queue...")

; Perform search on diary queue and then remove the STATUS field that
; it gives us for free

DialogSetItem dlg "diarymsg" msg1
Assign buffer (send opm :OP_SEARCH tmp1 searchattrs)

; Get the edit fields and save them on the dialog

Send dlg :OBJ_SETPROP :SEARCH_CRITERIA (GetEditFields dl)

DoList (fld buffer)

  Assign buff2 (Append buff2 [(NthRest 2 fld)])

EndDoList

Unless (First buff2)
;
; Nothing in queue for this user (or surrogate)

```

```

;

    SetBuffer dlg nil
    RefreshDataList dl
    DialogSetItem dlg "diarymsg" msg2
    Return nil

EndUnless

;
; Initialise dialog datalist with results of queue search
;
    Send dl :DL_S_BUFFER buff2

;
; Need to save the unchanged buffer as well as we will need this
; when using :OP_OPEN to open a picked abstract
;
    Send dl :OBJ_SETPROP :DLBUFF buffer
    RefreshDataList dl

    DialogSetItem dlg "diarymsg" ""

EndScript

;*****
;
; APGetDiaryOPEN:
; Called from OPEN button and double click on the Get From Diary dialog.
; Opens the picked folder into the environment
;
; ARGUMENTS: ignore dlg ignore
; RETURNS:  n/a
;*****
DefScript DYGetDiaryOPEN

    Arguments ignore dlg ignore

```

Locals picked

DialogSetItem dlg "diarymsg" ""

Assign picked (Send (GetDataList dlg) :DL_Q_PICK)

Unless picked

DialogSetItem dlg "diarymsg" "Please select a folder to open"

Return nil

EndUnless

When (> (Length picked) 1)

DialogSetItem dlg "diarymsg" "You may only open one folder at a time"

Return nil

EndWhen

ReturnFromPopup t

EndScript

;*****

; Clear the search criteria from the datalist

;*****

DefScript DYGetDiaryClear

Arguments ignore dlg ignore

Locals (env (GetCurrentEnvir)) dl dlsearch

DialogSetItem dlg "diarymsg" ""

Assign dl (GetDataList dlg)

Assign dlsearch (Send dlg :OBJ_GETPROP :SEARCH_CRITERIA)

When dlsearch

DoList (crit dlsearch)

Unless (Equal (First crit) "BRTEAM")

```

        Send dl :DL_S_STRING (First crit) ""
    EndUnless
EndDoList
EndWhen
Send dlg :OBJ_SETPROP :SEARCH_CRITERIA (GetEditFields dl)

SetBuffer dlg nil

RefreshDataList dl

Return

EndScript

;*****
; Forward many folders from the diary. Saves time opening and then forwarding
;*****
DefScript DYQuickForwardFromDiary

    Use "ROUTE"

    Arguments ignore dlg ignore

    Return [(Message ["Forward is currently not available."])]
    Locals opm datalist buffer picked fabs diariedfrom invoice errorlist tmplt total ~
        vend data databuf

    DialogSetItem dlg "diarymsg" ""

; Get the diary queue open manager

Unless (Assign opm (CSGetVal :BRQDIARY [[:TOPIC :OPM]]))
    Assign opm (CreateDevice "QUEUE")
    Send opm :OP_GOTO "BRQDRY"
    CSPutVal :BRQDIARY opm [[:TOPIC :OPM]]
EndUnless

; Get the picked list from the access manager

```

```
Assign datalist (GetDataList dlg)
Assign databuf (Send datalist :DL_Q_BUFFER)
Assign buffer (send datalist :OBJ_GETPROP :DLBUFF)
Assign picked (Send datalist :DL_Q_PICK)
```

```
Unless picked
  DialogSetItem dlg "diarymsg" "Please select one or more folders before forwarding"
  Return
EndUnless
```

```
; Loop through the selection and open each folder in turn
```

```
Assign total (Length picked)
```

```
DoTimes (num total)
```

```
Assign data (Nth (Nth num picked) databuf)
Assign tmplt (Send datalist :DL_Q_TEMPLATE)
Assign data (Send tmplt :TMPLT_BUF2ATTRS data)
```

```
Assign invoice (Lookup data "INV_NUM" 2)
Assign invoice (Or (StringTrim invoice) "None")
Assign vend (Lookup data "VEND_NAME" 2)
```

```
DialogSetItem dlg "diarymsg" (MakeString "Forwarding vendor: " vend ", invoice: " invoice " " (1+ num) " of "
total)
```

```
Assign fabs (First (Send opm :OP_OPEN buffer (Nth num picked) '@EDIT))
```

```
If fabs Then
```

```
; Get the diaried from property on the folder
```

```
Assign diariedfrom (Or (Send fabs :OBJ_GETPROP :DIARIED_FROM) "Unknown")
```

```
; If the folder was diarised from any queue other than invoices and MSA Keying
; then save the folder back to the diary queue and add the folder to the reject list
```

```

    If (Or (StringEqual diariedfrom "BRQINV") (StringEqual diariedfrom "BRQMSAK")) Then
;      Set the designated user on the target folder

      SetDocumentInfo fabs :ATTRIBS [["DESG_USER" (First (WhoAmI))]]

;      Set the action property to forward.

      Send fabs :OBJ_SETPROP :ACTION :FORWARD

;      Forward the document to the next task on its route

      DYRefAddEntry fabs "Quick forward from diary"
      RouteDocument fabs
;      Send opm :OP_SAVE fabs
    Else
      Push [(MakeString "Vendor: " vend ",Inv: " invoice ", Loc: " diariedfrom)] errorlist
      Send opm :OP_SAVE fabs
    EndIf

  Else

;    If the folder could not be opened then add the invoice number to the failed list

      Push (MakeString "Vendor: " vend ", Invoice: " invoice ", Loc: Locked") errorlist

    EndIf

  EndDoTimes

; After all successful folders have been forwarded, display a warning about those
; that were not.

DialogSetItem dlg "diarymsg" "Forward Complete"

When errorlist
  Message ["Invoices can only be forwarded in this manner if they were" ~
    "diarised from either MSA Keying or Invoices. The following" ~

```

```

        "invoices could not be forwarded:" "" "{errorlist}"]
EndWhen

DYGetDiarySEARCH nil dlg nil

DialogSetItem dlg "diarymsg" ""

Return t

EndScript

;*****
; Clear the search criteria from the datalist
;*****
DefScript DYGetDiaryCancel

    Arguments ignore dlg ignore

    Locals dl

    ; Get the edit fields and save them on the dialog

    Assign dl (GetDataList dlg)

    Send dlg :OBJ_SETPROP :SEARCH_CRITERIA (GetEditFields dl)

    ReturnFromPopup nil

EndScript

;*****
;
; Script Task on Diary Map which resets the Completion/Target date on a folder
; pulled out of the diary by a user.
;
; NOTE: (24-May-2006 Tedy)
; When a folder comes out of Diary because it had mail matched to new mail,
; the folder's target date is set to next working day, as follows:

```

```

; if today is Friday    Target Date is Today + 3 days,
;                      Saturday Target Date is Today + 2 days,
;                      otherwise Target Date is Today + 1 day only (next working day)
;
; Arguments newfabs    - folder abstract
;                  ignore    - folder request object
;
; Returns 0
;
;*****
DefScript DYDiaryResetTargetDate
Arguments fabs reqobj
Locals (env (GetCurrentEnvir)) wktype wtlookup wktypedata ~
      tdays tdreqid tdcurr tdnnext tdwtype tdtdays tdtdate ~
      vsNextWorkingDay DateNextWorkingDay WeekDayToday ~
      (vsToday (GetCurrentDateTime)) reqID

; Load necessary script files

; Unless (And ( fboundp 'BRCOMMsg ) ( fboundp 'BRCOMAddMemo ) ) ;AB1.1 - Commented
Out
  Unless (And ( fboundp 'BRCOMMsg ) ( fboundp 'BRCOMAddMemo ) (fboundp 'BRCOMTargetDate)) ;AB1.1 - Added
    LoadScript ( VSRootAppend "READDATA\\PROJECTS\\BR\\SCRIPT\\BRCOM.VSS" ) :useful
  EndUnless

  Unless ( fboundp 'BRSQLExecSP )
    LoadScript ( VSRootAppend "READDATA\\PROJECTS\\BR\\SCRIPT\\BRSQLE.VSS" ) :useful
  EndUnless

  Unless ( And ( fboundp 'CSPutVal ) ( fboundp 'CSGetVal ) )
;    LoadScript (VSRootAppend "PROJECTS\\COMMON\\CSDAT00L.VSS") :useful ; BW9 Commented Out
    LoadScript (VSRootAppend "PROJECTS\\COMMON\\CSDAT00L.FSL") :useful ; BW9 Added
  EndUnless

; *****
; Logging for Target date issues - DATESTRING, TIMESTRING, REQID, CURRENT, NEXT, WTYPE, TDAYS, TDATE
; *****

```

```

; Locals tdreqid tdcurr tdnnext tdwtype tdt days tdtdate

Assign tdreqid ( GetDocumentInfo fabs :field "REQID"      )
Assign tdcurr  ( GetDocumentInfo fabs :CURRENT           )
Assign tdnnext ( GetDocumentInfo fabs :NEXT              )
Assign tdwtype ( GetDocumentInfo fabs :field "BRWKTYPE"  )
Assign tdt days ( GetDocumentInfo fabs :field "BRTARGET" )
Assign tdtdate ( BComDateTimeString ( GetDocumentInfo fabs :field "BRTDATE" ) )

AppendToFile "K:\\P\\LOG\\TDATE\\TDATE.LOG" ( MakeString (DateString) "," (TimeString) "," tdreqid "," tdcurr
"," tdnnext "," tdwtype "," tdt days "," tdtdate "," "DYDiaryResetTargetDate - Start" )

; *****
; End of logging
; *****

;; ===== TEDY - 24 May 2006 ... =====
;; When a folder comes out of Diary because it has mail matched
;; to new mail, the folder's target date is set to next business day,
;; that is: if today is Friday add 3 days, Saturday add 2 days,
;; otherwise add one day only!
If (Equal (Send fabs :OBJ_GETPROP :CURRENT) :MAILMATCH)
  Then
    ; Get today's day of the week
    Assign WeekDayToday (FormatDateTime vsToday "ddd" nil)

    ; If it is Friday today add three (3) days to target date
    ; If it is Saturday add two (2) days to target date,
    ; For any other day of the week add just one (1) day!
  Cond
    Case (StringEqual WeekDayToday "Fri")
      Assign vsNextWorkingDay (DateTime+ vsToday 3 :Day)
    EndCase

    Case (StringEqual WeekDayToday "Sat")
      Assign vsNextWorkingDay (DateTime+ vsToday 2 :Day)
    EndCase

```

```

        Otherwise
            Assign vsNextWorkingDay (DateTime+ vsToday 1 :Day)
        EndCase
    EndCond
    Assign DateNextWorkingDay (FormatDateTime vsNextWorkingDay "dd/MM/yyyy" nil)
    SetDocumentInfo fabs :attribs [{"BRTDATE" DateNextWorkingDay}]

    ;;--- Display a message on the Process Agent...
    Assign reqID (GetDocumentInfo fabs :field "REQID")
    BRComMsg (MakeString "Mail-Matched folder REQID: " reqID " - Target Date set to next working day: "
DateNextWorkingDay) "BRCOM.VSS" "BRCOMInitWPTracking" 2

    ;; Add entry to TDATE.LOG just for debug purposes...
    AppendToFile "K:\\P\\LOG\\TDATE\\TDATE.LOG" ( MakeString (DateString) "," (TimeString) "," tdreqid "," tdcurr
    "," tdnnext "," tdwtype "," tdtdays "," tdtdate "," "DYDiaryResetTargetDate - Target Date set to next work day!" )

Else
;; ===== ... TEDY - 24 May 2006 =====

; If the next location is :DIARY then don't mess with the target date.
; This is ok as to have got here the folder must have been opened from the diary queue and forwarded
; by a user. This should have set the next location to something other than :DIARY unless they are really
; sending it back. If they are sending it back then the target date should have been reworked by the
; diary setup functions.

Unless ( StringEqual ( GetDocumentInfo fabs :NEXT ) :DIARY )

    Assign wktype ( GetDocumentInfo fabs :field "BRWKTYPE" )

    Unless ( Assign wtlookup ( CSGetVal :BRWORKTYPES2 [ [ :TARGET env ] [ :TOPIC :CONFIG_VALS ] ] ) )
        Assign wtlookup ( BRSQLExecSP "sp_brsselwtnotfirst" )
        CSPutVal :BRWORKTYPES2 wtlookup [ [:TARGET env ] [ :TOPIC :CONFIG_VALS ] ]
    EndUnless

    Assign wktypedata ( Lookup wtlookup wktype )
    Assign tdays ( Nth 5 wktypedata )

```

```

        ;Reset the target date on the folder to ( Current Date + Target days for the folder's worktype)

        ;SetDocumentInfo fabs :attribs [ [ "BRTDATE" ( DateTime+ ( GetCurrentDateTime ) tdays :day ) ] ] ;AB1.1
- Commented Out
        SetDocumentInfo fabs :attribs [ [ "BRTDATE" ( BRCOMTargetDate ( GetCurrentDateTime ) tdays ) ] ] ;AB1.1 -
Added

        EndUnless

;; ===== TEDY - 24 May 2006 ... =====
        EndIf
;; ===== ... TEDY - 24 May 2006 =====

; *****
; Logging for Target date issues - DATESTRING, TIMESTRING, REQID, CURRENT, NEXT, WTYPE, TDAYS, TDATE
; *****

; Locals tdreqid tdcurr tdnnext tdwtype tdtdays tdtdate

        Assign tdreqid ( GetDocumentInfo fabs :field "REQID"      )
        Assign tdcurr  ( GetDocumentInfo fabs :CURRENT           )
        Assign tdnnext ( GetDocumentInfo fabs :NEXT              )
        Assign tdwtype ( GetDocumentInfo fabs :field "BRWKTYPE"  )
        Assign tdtdays ( GetDocumentInfo fabs :field "BRTARGET" )
        Assign tdtdate ( BRCOMDateTimeString ( GetDocumentInfo fabs :field "BRTDATE" ) )

        AppendToFile "K:\\P\\LOG\\TDATE\\TDATE.LOG" ( MakeString (DateString) "," (TimeString) "," tdreqid "," tdcurr
        "," tdnnext "," tdwtype "," tdtdays "," tdtdate "," "DYDiaryResetTargetDate - End" )

; *****
; End of logging
; *****

        Return 0

EndScript

```

```

;=====
=====
;
; Author:      Mayank Ladd
; Version:     BW12
; Date:        06/05/1997
;
; Description: Script Task on Diary Map which resets document assessment dates.
;              Folder expires from the diary to workbench and has been in the
;              diary for greater than 24 Hours.
;
; Returns:     0
;
;=====
=====

```

DefScript DYDiaryResetDocAssStartDate

 ;BW12 ADDED

 Arguments fabs reqobj

 ;BW12 ADDED

 Locals docsbm vdoc newdate olddate doctype reqid (dtcount 0)

 ;BW12 ADDED

 Unless (fboundp 'BrComMirTrkDoc)

 ;BW12 ADDED

 LoadScript (VSRootAppend "READDATA\\PROJECTS\\BR\\SCRIPT\\BRCOM.VSS") :useful

 ;BW12 ADDED

 EndUnless

 ;BW12 ADDED

 Unless fabs

 ;BW12 ADDED

 BRCOMMMsg (MakeString "No folder abstract passed in as an argument.") "DYDIARY.VSS"

"DYDiaryResetDocAssStartDate" 1 ;BW12 ADDED

 Return nil

 ;BW12 ADDED

```

EndUnless
    ;BW12 ADDED

Assign docsbm (GetDocumentInfo fabs :BR_DOCS_BEFORE_M_MATCH)
    ;BW12 ADDED

Assign reqid (Send reqobj :REQ_Q_ID)
    ;BW12 ADDED

;=====
    ;BW12 ADDED
; If folder expires from diary and has been in the diary for longer
    ;BW12 ADDED
; than a day then reset assessment date of documents.
    ;BW12 ADDED
;
    ;BW12 ADDED
; Note. If the folder has been matched then we do not reset the date
    ;BW12 ADDED
; on those that have just matched. (If docsbm...). The property
    ;BW12 ADDED
; :BR_DOCS_BEFORE_M_MATCH is set in mail matching is the number
    ;BW12 ADDED
; of documents in the target folder before it was matched.
    ;BW12 ADDED
;=====
    ;BW12 ADDED

BRCommMsg (MakeString "Folder Diary date " (GetDocumentInfo fabs :field "BRDIARYIN") ~
    ;BW12 ADDED
    " Sent to diary before Today? " (< (datetime- (GetDocumentInfo fabs :field "BRDIARYIN")
(GetCurrentDateTime) :day ) 0)~ ;BW12 ADDED
    " Reqid " reqid ) ~
    ;BW12 ADDED
    "DYDIARY.VSS" "DYDiaryResetDocAssStartDate" 1
    ;BW12 ADDED

If (< (datetime- (GetDocumentInfo fabs :field "BRDIARYIN") (GetCurrentDateTime) :day ) 0)

```

Then

```
;BW12 ADDED
WithOpenDocument (fvdoc fabs)
;BW12 ADDED
; If documents have just been matched we ignore them by setting the dtcount to the number of
;BW12 ADDED
; documents in the folder before it was mail matched. Otherwise we reset all document
;BW12 ADDED
; assessment dates.
;BW12 ADDED
If docsbm Then
;BW12 ADDED
    Assign dtcount docsbm
;BW12 ADDED
    Send fvdoc :OBJ_REMPROP :BR_DOCS_BEFORE_M_MATCH
;BW12 ADDED
Else
;BW12 ADDED
    Assign dtcount (TotalFolderDocs fvdoc)
;BW12 ADDED
EndIf
;BW12 ADDED

DoTimes (cnt dtcount)
;BW12 ADDED
    Assign vdoc (First (SelectFromFolder fvdoc :itemnums [(+ cnt 1)]))
;BW12 ADDED
    If vdoc
;BW12 ADDED
        Then
;BW12 ADDED
            Assign olddate (GetDocumentInfo vdoc :field "BRASSTART")
;BW12 ADDED
            Assign newdate (BrComMirTrkDoc "DYDiaryResetDocAssStartDate" vdoc fabs)
;BW12 ADDED
            Assign doctype (GetDocumentInfo vdoc :field "BRDOCTYPE")
;BW12 ADDED
            SetDocumentInfo vdoc :attribs [{"BRASSTART" newdate}]
```

```

;BW12 ADDED
    BRCommMsg (MakeString "Assessment date reset from " olddate " to " newdate " on document " cnt "
doctype " doctype) "DYDIARY.VSS" "DYDiaryResetDocAssStartDate" 1 ;BW12 ADDED
    Else
;BW12 ADDED
    BRCommMsg (MakeString "Error - Failed to reset assessment date on document " cnt " in folder "
reqid) "DYDIARY.VSS" "DYDiaryResetDocAssStartDate" 1 ;BW12 ADDED
    Endif
;BW12 ADDED
    EndDoTimes
;BW12 ADDED

```

```

    BRCommMsg (MakeString "Number of documents reset " dtcount " out of " (TotalFolderDocs fvdoc))
"DYDIARY.VSS" "DYDiaryResetDocAssStartDate" 1 ;BW12 ADDED

```

```

EndWithOpenDocument
;BW12 ADDED

```

```

EndIf
;BW12 ADDED

```

```

Return 0
;BW12 ADDED

```

```

EndScript
;BW12 ADDED

```

```

;; =====
;; TEDY - Nov 2004 ...

;; TEDY - July 2006 ... Added section DiarySpecial and entry LimitWorkTypes, in
;; order to achieve same behaviour for diarising a list of
;; worktypes (DaysInDiaryMax) as done for the new HB Claims.

;-----
; BRGetDiaryParam
;

```

```

; DESCRIPTION:
;   Get initialization stuff from ini file, or if repetitely called,
;   use same of the environment (probably working faster that way?)
;
; ARGUMENTS:
;   Section - string as found enclosed in square brackets in ini file
;   Entry   - string as found followed by '=' in the given section.
;
; OPTIONALS:
;   nosave - t - if config should be read from file and not set on env.
;
; RETURNS:
;   nil - if INI file not found, or
;   value found in Section / Entry in INI file.
;-----
DefScript BRGetDiaryParam
Arguments Section Entry
Optionals nosave
Locals    iniFile (env (GetCurrentEnvir)) ret keywSection keywEntry ~
            (sectionsList [["HBReminderLetter" :DIARY] ["DiarySpecial" :DIARY]]) ~
            (entriesList [
                ["DaysWait"      :DAYSWAIT ] ~
                ["DaysINPUT"     :DAYSINPUT] ~
                ["DaysOUT"       :DAYSOUT  ] ~
                ["DaysInDiaryMax" :DAYSINDIARYMAX] ~
                ["WorkTypes"     :NEWCLAIMSWKTYPES] ~
                ["LimitWorkTypes" :LIMITWKTYPES] ~
                ["Managers"      :MANAGERS] ~
            ]
        )

Unless (FBoundP 'CSGetVal)
    LoadScript (VSRootAppend "projects\\common\\csdatool.fsl") :mandatory
EndUnless

Assign iniFile (CSGetVal :DIARYINIFILE [[:TARGET env] [:TOPIC :DIARY]])
Unless iniFile
    Assign iniFile (VSRootAppend "vs_bin\\brcom.ini")

```

```

Unless (ProbeFile iniFile)
    Return nil
EndUnless

;; -- Store full path and file name of INI, on env...
Unless nosave
    CSPutVal :DIARYINIFILE iniFile [[:TARGET env] [:TOPIC :DIARY]]
EndUnless
EndUnless

;; -- Case insensitive lookup...
Assign keywSection (Lookup sectionsList Section 2 :test 'StringEqual)
Assign keywEntry   (Lookup entriesList Entry 2 :test 'StringEqual)

Assign ret (CSGetVal keywEntry [[:TARGET env] [:TOPIC keywSection]])
Unless ret
    Assign ret (ReadFromString (GetConfigString iniFile Section Entry))
    Unless nosave
        CSPutVal keywEntry ret [[:TARGET env] [:TOPIC keywSection]]
    EndUnless
EndUnless

Return ret

EndScript ;BRGetDiaryParam

```

```

;-----
; BRNewAppsRules
;   Conditional code for HB - New claims
;
; ARGUMENTS:
;   dlg      - dialog object
;   fabs     - folder object
;
; RETURNS
;   t        - if diary setting ok
;   nil      - if diary settings are invalid

```

```

;
; NOTE:
;   daysInDiaryMax is set to 29 days in BRCOM.INI
;
;-----
DefScript BRNewAppsRules
Arguments dlg fabs
Locals    diaryIN diaryOUT vsdiaryIN vsdiaryOUT (daysInDiary 0) ~
          (user (First(WhoAmI))) managersList daysInDiaryMax

;; -- Get list of privileged users from INI file...
Assign managersList (BRGetDiaryParam "HBReminderLetter" "Managers")

Send (Send dlg :DIALOG_Q_CTRL "cbox2") :WND_S_ENABLED :NO
Send (Send dlg :DIALOG_Q_CTRL "actions") :WND_S_ENABLED :NO

;; -- No manual overtype the Diary reason (selected from combo box)...
Send (Send dlg :DIALOG_Q_CTRL "res") :WND_S_ENABLED :NO

If (Member user managersList)
  Then
    ;; -- Enable controls...
    BREnableDiaryCtrl dlg
    SetProcessTitle dlg " Diary Setup - Team Leader"
  Else
    SetProcessTitle dlg " Diary Setup"
    ;; -- Get Diary dates, IN and OUT
    Assign diaryIN (GetDocumentInfo fabs :field "BRDIARYIN")
    Assign diaryOUT (GetDocumentInfo fabs :field "BRDIARYOUT")
    When diaryIN
      Assign daysInDiary (DateTime- diaryOUT diaryIN :day)
    EndWhen

    ;; -- Convert dates to VS format so we can calculate difference...
    ;When diaryIN
    ; Assign vsdiaryIN (ParseDateTime diaryIN "dd/MM/yyyy" nil)
    ;EndWhen
    ;When diaryOUT

```

```

; Assign vsdiaryOUT (ParseDateTime diaryOUT "dd/MM/yyyy" nil)
;EndWhen

Assign daysInDiaryMax (BRGetDiaryParam "HBReminderLetter" "DaysInDiaryMax")

If (>= daysInDiary daysInDiaryMax)
  Then
    ;; -- Disable controls...
    BRDisableDiaryCtrl dlg
    Send fabs :OBJ_SETPROP :ALLOWUSEROK t
  Else
    ;; -- Enable controls...
    BREnableDiaryCtrl dlg
  EndIf
EndIf

EndScript ;BRNewAppsRules

```

```

;-----
; BRDiaryOKRule
;   Conditional code for HB-New claims (listed in HBReminderLetter section
;   of BRCOM.INI - also containing DaysInDiaryMax parameter.
;   Only team leaders are able to diarise beyond the original DaysInDiaryMax
;   (set to 29 days in BRCOM.INI).
;   Cases that have already been sent to diary beyond this max period (that
;   is the diary count is more than 0, are ok to be returned to diary for
;   that return date, by officers. Extending that date can only be done by
;   a team leader and to a maximum of DaysInDiaryMax as of today.
;
; ARGUMENTS:
;   abs      - folder object
;   dlg      - dialog object
;   retdate  - return date from diary
;   rettime  - return time from diary
;
; RETURNS
;   t        - if diary setting ok

```

```

;    nil - if diary settings are invalid
;-----
DefScript BRDiaryOKRule
Arguments abs dlg retdate rettime
Locals    workType listHBNewApps diaryIN vsdiaryIN      ~
          vsretdate diaryOUTmax managersList allowUserOK ~
          (user (First(WhoAmI))) vsdiaryOUTmax daysInDiaryMax ~
          (today (GetCurrentDateTime))

;; -- If folder has never been sent to Diary, set the diaryin date to today's date...
Assign vsdiaryIN (GetDocumentInfo abs :field "BRDIARYIN")

;; -- Convert dates to VS format to calculate difference in days...
Assign vsretdate (ParseDateTime retdate "dd/MM/yyyy" nil)
Unless vsretdate
    ;; -- Ensure that the return field is not left empty...
    DialogSetItem dlg "msg" (MakeString "Please set a diary return date!")
    SYS::BEEP
    Return nil
EndUnless

If (< (DateTime- vsretdate today :day) 0)
    Then
        ;; -- Tell that date is invalid...
        DialogSetItem dlg "msg" (MakeString "Return date must be between " today " and " diaryOUTmax)
        SYS::BEEP
        Return nil
    EndIf

;; -- Get list of privileged users from INI file...
Assign managersList (BRGetDiaryParam "HBReminderLetter" "Managers")

;; -- Get days max in diary (or default to 29 if INI entry not found)...
Assign daysInDiaryMax (BRGetDiaryParam "HBReminderLetter" "DaysInDiaryMax")
;; -- If not found in INI file set to default of 29 days...
Unless daysInDiaryMax
    Assign daysInDiaryMax 29
EndUnless

```

```

If (And managersList (Member user managersList))
  Then
    ;; -- For Managers - allow 29 days from today...
    Assign vsdiaryOUTmax (DateTime+ today daysInDiaryMax :day)
    If (> (DateTime- vsretdate today :day) daysInDiaryMax)
      Then
        Assign diaryOUTmax (FormatDateTime vsdiaryOUTmax "dd/MM/yyyy" nil)
        DialogSetItem dlg "msg" (MakeString "Return date may not be later than " diaryOUTmax)
        SYS::BEEP
        Return nil
      Else
        Send abs :OBJ_SETPROP :ALLOWUSEROK t
      EndIf
    Else
      ;; -- Where the diary setup is disabled, a property that is set on the folder
      ;;   allows an officer to send the folder to Diary, even if the return date
      ;;   which is grayed out is greater than the max of 29 days
      ;;   NOTE: This is especially useful for the interim situation, where a large
      ;;         number of folders have already been sent to diary for periods that
      ;;         exceed the 29 days. Also applies to folders that the team leader
      ;;         had set a return that extended the period beyond 29 days and an
      ;;         officer opens a folder from diary and wishes to return it to diary.
      Assign allowUserOK (Send abs :OBJ_GETPROP :ALLOWUSEROK)
      When allowUserOK
        Return t
      EndWhen

      ;; -- For Officers - allow 29 max, unless a team leader has already set it higher...
      Unless vsDiaryIN
        ;; -- Folder has not been to Diary before,
        ;;   so today's date would be the diary in date...
        Assign vsDiaryIN today
      EndUnless
      If (> (DateTime- vsretdate vsDiaryIN :day) daysInDiaryMax)
        Then
          ;; -- Tell that only a team leader could do this...
          Assign diaryOUTmax (FormatDateTime (DateTime+ vsdiaryIN daysInDiaryMax :day) "dd/MM/yyyy" nil)

```

```

        DialogSetItem dlg "msg" (MakeString "Please refer to your Team leader to diarise beyond " diaryOUTmax)
        SYS::BEEP
        Return nil
    EndIf
EndIf
;; -- Will send this folder to Diary...
;Inspect ["OK to send to Diary" "vsDiaryIN vsretdate" vsDiaryIN vsretdate]
Return t

```

```
EndScript ;BRDiaryOKRule
```

```

;-----
;  BREnableDiaryCtrl
;    Enable controld on the diary dialog
;
;  ARGUMENTS:
;    dlg - dialog object
;
;  RETURNS
;    n/a
;-----

```

```
DefScript BREnableDiaryCtrl
```

```
Arguments dlg
```

```

        Send (Send dlg :DIALOG_Q_CTRL "retdate") :WND_S_ENABLED :YES
        Send (Send dlg :DIALOG_Q_CTRL "time") :WND_S_ENABLED :YES
        Send (Send dlg :DIALOG_Q_CTRL "count") :WND_S_ENABLED :YES
        Send (Send dlg :DIALOG_Q_CTRL "mag") :WND_S_ENABLED :YES
        Send (Send dlg :DIALOG_Q_CTRL "cbox") :WND_S_ENABLED :YES
EndScript ;BREnableDiaryCtrl

```

```

;-----
;  BRDisableDiaryCtrl
;    Enable controld on the diary dialog
;
;  ARGUMENTS:

```

```

;   dlg - dialog object
;
;   RETURNS
;   n/a
;-----
DefScript BRDisableDiaryCtrl
Arguments dlg

    Send (Send dlg :DIALOG_Q_CTRL "retdate") :WND_S_ENABLED :NO
    Send (Send dlg :DIALOG_Q_CTRL "time") :WND_S_ENABLED :NO
    Send (Send dlg :DIALOG_Q_CTRL "count") :WND_S_ENABLED :NO
    Send (Send dlg :DIALOG_Q_CTRL "mag") :WND_S_ENABLED :NO
    Send (Send dlg :DIALOG_Q_CTRL "cbox") :WND_S_ENABLED :NO

EndScript ;BRDisableDiaryCtrl

;; ... TEDY - Nov 2004
;; =====

```