```
;===========================================================================
;    EEEEE  DDDDD   SSSSS
;    EE     DD  DD  SS
;    EEEE   DD  DD  SSSSS
;    EE     DD  DD     SS  (c) Copyright 1996 Electronic Data Systems Ltd.
;    EEEEE  DDDDD   SSSSS      All rights reserved
;===========================================================================
;
;  File     : <readdata>\PROJECTS\BR\BRCOM.VSS
;  Author   : S Burgoyne
;  Date     : 1st Jan 96
;  Function : Commonly used functions.
;
;===========================================================================
;
;  CCS         : 48                    Author : Simon Burgoyne
;  Date    : 17th june 1996
;  Version     : 1.1
;  Description : Chanage open method now that folder type and stream combo
;                box's have been removed.
;
;===========================================================================
;
;  CCS         : 53                    Author : Ajay Bhatt
;  Date    : 19th june 1996
;  Version     : 1.2
;  Description : Added 'BRWBATTRS' to BRCOMBUTON function
;
;===========================================================================
;
;  CCS         : 57                    Author : Ajay Bhatt
;  Date    : 25th june 1996
;  Version     : 1.3
;  Description : Added function BRCOMTargetDate
;
;===========================================================================
;
;  CCS         : 58                Author : Ajay Bhatt
```

```
;  Date    : 25th june 1996
;  Version     : 1.4
;  Description : Set open time on newly created folders so that they
;                are tracked in the BRUSER table which records work done by users
;                Also, only set last user attributes on a folder when forwarded by
;                a real user NOT a Process Agent.
;
;============================================================================
;
;  CCS         : 61                Author : Ajay Bhatt
;  Date    : 12th june 1996
;  Version     : 1.5
;  Description : BRCOMTargetDate needs to take into account the from_date
;                falling on either a Saturday or Sunday
;
;============================================================================
;
;  CCS         : 71                    Author : M Ladd
;  Date    : 06 August 1996
;  Version     : 1.6
;  Description : Refer dialog from commissioner includes all teams and their users
;
;============================================================================
;
;  CCS         :                       Author : A Bhatt
;  Date    : 17 August 1996
;  Version     : 1.7
;  Description : Initialise the Archive Date to empty string for Auto_indexed docs.
;
;============================================================================
;  CCS         : 83             Author : A Bhatt
;  Date    : 03 September 1996
;  Version     : 1.8
;  Description : Select Work Description to use in open method for
;                certain worktypes
;
;============================================================================
;
```

```
;  Project  : Wandsworth
;  CCS      : BW1
;  Author   : Mayank Ladd
;  Date     : March 97
;  Version  : Y2000
;  Desc     : Call date functions from a script function.
;             BRComDateTimeString    - Outputs Date and Time as a string
;             BRComInternalDateTime  - Converts Date and Time into viewstar internal format
;             BRComDateString        - Outputs Date Only as a string
;             BRComInternalDate      - Converts Date into viewstar internal format
;
;===============================================================================
;
;  Project     : Wandsworth
;  CCS         : BW6
;  Author      : Mayank Ladd
;  Date    : 07/04/97
;  Version     :
;  Description : Wandsworth Conversion - Remove LGVision Interface
;
;===============================================================================
;
;  Project  : Wandsworth
;  CCS      : BW9
;  Author   : Mayank Ladd
;  Date     : 15 April 1997
;  Version  :
;  Desc     : Script \PROJECTS\COMMON\CSDATOOL.VSS not present, loading CSDATOOL.FSL
;
;===============================================================================
;
;  Project  : Wandsworth
;  CCS      : BW12
;  Author   : Mayank Ladd
;  Date     : 02 May 1997
;  Version  :
;  Desc     : Track Documents within Folders.  Added to track start date for a document
;
```

```
;=============================================================================
;
;  Project     : Wandsworth
;  CCS         : BW15
;  Author      : Simon Burgoyne
;  Date    : 1/5/97
;  Version     :
;  Description : Wandsworth Conversion - Commissioner to Client Officer.
;
;=============================================================================
;
;  Project  : Wandsworth
;  CCS      : CCS 4
;  Author   : Simon Burgoyne
;  Date     : 17th June 1997
;  Version  :
;  Desc     : Temporarily comment out update display on del page and del doc functions
;           : we are expecting a fix from VS here VS call no. RF6472 EDS No. 1035
;
;=============================================================================
;
;  Project  : Wandsworth
;  CCS      : 11
;  Author   : Roger C. King
;  Date     : 3 July 1997
;  Version  :
;  Desc     : Change to BRCOMSetSearchAttrs. Added change to enable/disable PRIORITY
;             in the open method search_criteria and sort_criteria.
;
;=============================================================================
;
;  Project  : Wandsworth
;  CCS      : 12 (HelpDesk Call 1190)
;  Author   : Andy Bell
;  Date     : 10 June 1998
;  Version  :
;  Desc     : Change to BRCOMCreateFolder
;             Pick for Worktype now selects NIL rather than the first item in the list
```

```
;             so the user must select a WorkType
;
;===============================================================================
;
;  Project  : Wandsworth
;  Author   : Tedy Shalev (for Mosaix Ltd)
;  Date     : 9 March 1999
;  Version  :
;  Desc     : Work distribution restructure
;             Add new script function BRSetUserSearchCriteria.
;             Set a section in CONFIG_VALS determining whether or not the current
;             user is allowed access to the "Count Folders..." option.
;             Modified icons for Yes and No buttons in dialog box displayed by
;             the BRCOMShowMessageBox script function.
;             The Next button sets the search_criteria for OpenFolder to include
;             a UPRN Range, if one is allocated (colour based groups only.)
;
;             Fixed bug where using "In Target" sorting and clicking  "Next"
;             (refer BRCOMSetSearchCriteria) date has to be MM/dd/yyyy format!
;             Added an optional parameter to the BRCOMSetSearchCriteria script,
;             to handle the setting of Process Mode criteria, with priority and
;             In Target sorting.
;
;===============================================================================
;
;  Project  : Wandsworth
;  Author   : Tedy Shalev (for Mosaix Ltd)
;  Date     : 28 October 1999
;  Version  :
;  Desc     : BRCOMDelUserDocs updated - UpdateDisplay now working in 32 bit.
;             NOTE: The "MS Word..."  menu was added to the "Client Officer" appl,
;                   and so this script is also utilised in this application, as it
;                   it is in workbench.
;
;===============================================================================
;
;  Project  : Wandsworth
;  Author   : Tedy Shalev (for Mosaix Ltd)
```

```
;  Date     : 01 November 1999
;  Version  :
;  Desc     : Added batchIndexToArchive group of batches to support more than one
;             Batch that will Scan, Auto or Manual Index followed by Archive.
;
;===============================================================================
;
;  Project  : Wandsworth
;  Author   : Tedy Shalev (for Mosaix Ltd)
;  Date     : 17 November 1999
;  Version  :
;  Desc     : batchIndexToArchive group sets BRCUSERID to "AUTO".
;             If BRCUSERID is more than 10 chars, the Archive process will fail!
;
;===============================================================================
;  Project  : Wandsworth
;  Author   : Tedy Shalev (for Mosaix Ltd)
;  Date     : 03 October 2000
;  Version  :
;  Desc     : When Auto-Indexing, treat folders by Folder Type, ie.
;             BRFLDTYPE = 'R' (Revenue folders) send to either CTax1 or CTax2 team,
;             depending upon the UPRN division (defined in brcom.ini)
;             Only BRFLDTYPE = 'B' (Benefits folders) will be directed to Housing
;             Benefits Teams (Red, Green, White or Blue)
;
;             If Auto Indexing results in no Team and WorkGroup allocation, due to
;             UPRN that is not allocated to any of the HB Teams, folder will be
;             forwarded to Manual Indexing, purely for Team/Workgroup decision!
;===============================================================================
;
;  Author   : Tedy Shalev
;  Date     : 16 November 2004
;  Version  :
;  Desc     : Modified BRCOMDelUserDocs - better looking message displayed!
;             Only the original OWNER (who created the Word document) or the SYSADM
;             may delete a Word document that was added to a folder...
;===============================================================================
;
```

```
;  Author   : Tedy Shalev
;  Date     : 15 April 2005
;  Version  :
;  Desc     : Report for iWorld in CSV format (if document type is found) invoked
;             by BRiWorldCSVReport, called from BRCOMSetBRData and only if folder
;             is auto-indexed (that is property :BRIDXOK :YES is set on the folder)
;             Folders sent to Manual indexing will be reported on the BeforeForward
;             hook, using the same BRiWorldCSVReport script.
;============================================================================
;
;  Contents:
;  ---------
;  BR_Dummy_BRCOM
;  BRCOM_GetFldType            - Calculate folder worktype by ranking documents in folder.
;  BRCOMSetBRData              - Get indexing data from UPRN\Surname or NI number.
;  BRCOMAddMemo                - Add memo txt to folder and first doc of fabs passed or fabs open.

;  BRComForward                - Set :CURRENT AND :NEXT properties on an abstract and forward it.
;  BRCOMSetSearchCriteria
;  BRCOMSetupAppCustomData     - Sets up Custom data to be saved on application environments
;  BRComSetGroupUsers          - Sets up Group/User data to be saved on application environments
;  BRComMsg                    - Common message function
;  BRComPadString              - Pads a string to the specified length
;  BRComDateTimeString         - BW1 Checks type of date & time arguments and formats it into a string
;  BRComInternalDateTime       - BW1 Checks type of date & time arguments and converts into internal format
;  BRComDateString             - BW1 Checks type of date (only) arguments and formats it into a string
;  BRComInternalDate           - BW1 Checks type of date (only) arguments and converts into internal format
;  BRComAddCopy                - Adds copies of documents from View only app to the current folder in process
layout apps
;  BRComCreateNewFolder        - Creates a new folder and opens it in a forward environment
;  BRCOMHistMenu               - Called when the script is called from the WorkBench Application
;  BRCOMHistStatus             - Called when the user selects the Status Button from the View Folder History
Dialog
;  BRCOMHistHist               - Called when the user selects the History Button from the View Folder History
Dialog
;  BRCOMHistUsers              - Called when the user selects the Users Button from the View Folder History
;  BRCOMHistDisplayPopup       - Called from BRCOMHistMenu inorder to popup the dialog box in the Workbench
Application
```

```
;  BRCOMHistDisplayRows         - Called when BRCOMHistStatus, BRCOMHistHist, BRCOMHistUsers has data to
;                                 display as rows in the dialog application, under the Workbench
;                                 Application.
;  BRCOMButton                  - Called from key buttons within Workbench, Enquiry and Commissioner.
;  BRCOMHelp                    - Called from the environment - Context Sensitive Help file resident under the
Help Option in Various Appplications.
;
;  BRCOMPopup                   - Called by all occurances of the custom dialog window created in the ViewStar
Application.
;  BRCOMShowmessagebox          - Version of ShowMessageBox which calls BRCOMPopup which allows us to control the
positioning of message boxes as well as dialog boxes.
;  BRComLogEvent                - The routine will handle the renaming and deletion of log files when they grow
past a specified limit.
;  BRComLogExt                  - Function to create a three character file extension given a log file number
;  BRCOMTargetDate              - Calculates Target Dates given the Start date and the interval
;
;  Scripts/functions called (required preloads):
;  ---------------------------------------------
;  None
;
;  Objects loaded/saved:
;  ---------------------
;  None
;
;  Undocumented Viewstar considerations:
;  -------------------------------------
;  fldr_new             - the function called when the 'New'     option is selected from the 'Folder' Menu (a
Standard VS feature)
;  fldr_addcopy         - the function called when the 'AddCopy' option is selected from the 'Folder' Menu (a
Standard VS feature)
;  GetDocumentsFromView - Pre-check function for 'AddCopy' option of the'Folder' Menu (a Standard VS feature)
;  @string_decrypt      - lisp function to decrypt VS passwords
;  @string_encrypt      - lisp function to encrypt VS passwords
;
;=========================================================================
DefScript BR_Dummy_BRCOM                    ;Dummy function for BRCOMM.VS
;; BRComSetGroupUsers nil
   Return nil
```

```
EndScript


;=========================================================================
;
; Calculate folder worktype by ranking document types in the folder.
;
; When        Script Calls:    BRIMPORT.BRImportSetFldProps
;
; Arguments   fabs             Folder abstract to rank.
;
; Returns     Folder Type or nil
;
;=========================================================================
DefScript BRCOM_GetFldType                  ;Calculate folder worktype by ranking document types in the folder.

   Arguments fabs

   locals (env (GetCurrentEnvir)) (maxwt 0) (fldtype 0) (flddefwt 0) (docdefwt 0)~
          wktypes docuprn docsurn prop (cnt 0)

   ;Get the highest worktype and add 1
   Unless (Fboundp 'BRSQLExecSP)
     LoadScript (VSRootAppend "readdata\\projects\\br\\script\\brsql.vss") :infreq
   EndUnless

   ;Run through all documents in the folder to get information to populate folder with.
   WithOpenDocument (fvdoc fabs)
      DoList (vdoc (SelectFromFolder fvdoc :all t))
         Assign cnt (+ 1 cnt)
         ;1) Get the max default worktype value and use this to choose a folder
         ;   worktype to set.
         ;NB Make sure that the default doc type exists and is a number if not ignore it.
         If (And (Assign docdefwt (GetDocumentInfo vdoc :field "BRDEFWKTYPE"))~
                 (Type docdefwt '@INTEGER))~
         Then
;
;          Jump out of the loop when the first document with an associated worktype is found
```

```
;
           Assign flddefwt docdefwt
           Return
         Endif
      EndDoList
   EndWithOpenDocument

   ;If flddefwt is still the defualt value then no document types have been set so return nil.
   When (Equal flddefwt 0)
      Return nil
   EndWhen

   ;Get row from worktype table
   If (Assign fldtype (First(BRSQLExecSP "sp_brgetworktype" [flddefwt])))
   Then
      Return fldtype
   Else
      Return nil
   Endif

EndScript


;;=============================================================================
;;
;; Get Data from database abstract and update folder attributes.
;;
;; When        Called.
;;
;; Returns     NA
;;
;;=============================================================================
DefScript BRCOMSetBRData               ;Get indexing data from UPRN\Surname or NI number.
Arguments fabs reqobj
Optionals (selnames []) (distinct nil) (orderby [])
Locals    tmplt matchdata matchtype hbis ctax ni uprn surn attribs type flag rows    ~
          ck_attrs ck_tmplt ck_errs ck_data batch attribs2 brdattype_pos newPairList ~
          (ini_file (VSRootAppend "VS_BIN\\BRCOM.INI")) batchIndexToArchive ~
```

```
        team uprnCTax wkGrp


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

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

   Assign tmplt (Send '@namemgr_create :NAME_GET "TEMPLATE" "BRIDXDLG")
   Assign brdattype_pos (@position "BRDATTYPE" (Send tmplt :TMPLT_Q_ITEM_NAMES) :test '@equal)

   ;=====================
   ;Search DB for matches
   ;=====================

   ;Start the search on HBIS number

   When (Assign hbis (GetDocumentInfo fabs :field "BRHBISNO"))
      When (Assign matchdata (BRComIdxSearch [["BRHBISNO" hbis]]))
         BRComMsg (MakeString "Match on HBIS : " matchdata) "BRCOM.VSS" "BRCOMSetBRData" 2
         Assign type 1                                                    ;Get HBIS data we don't want CTAX
data
      EndWhen
   EndWhen

   ;Unless we have a match search on CTAX number

   Unless matchdata
      When (Assign ctax (GetDocumentInfo fabs :field "BRCTAXNO"))
         When (Assign matchdata (BRComIdxSearch [["BRCTAXNO" ctax]]))
            BRComMsg (MakeString "Match on CTAX : " matchdata)  "BRCOM.VSS" "BRCOMSetBRData" 2
            Assign type 0                                                ;Get CTAX data we don't want HBIS data
         EndWhen
      EndWhen
   EndUnless
```

```
   ;Unless we have a match search on National Insurance number

   Unless matchdata
      When (Assign ni (GetDocumentInfo fabs :field "BRNINO"))
         When (Assign matchdata (BRComIdxSearch [["BRNINO" ni]]))
           BRComMsg (MakeString "Match on NINO : " matchdata)  "BRCOM.VSS" "BRCOMSetBRData" 2
           Assign type 1                                                ;Get HBIS data we don't want CTAX data
         EndWhen
      EndWhen
   EndUnless

   ;=======================================
   ;Select a valid row from the row returned
   ;=======================================

   ; If a valid row cannot be found then route the folder to manual indexing (Return 0, do not set :BRIDXOK to
:YES).
   ; If we have more than one row returned then take the last row for the DB search type (CTAX = 0, HBIS = 1).

   ; NB No special logic for backfile documents!
   ; We must have a match for backfile folders as well. Or they will have to sent to manual indexing.
   ; We need the match to get the UPRN number. We get the following attribs from the import.
   ; "BRHBISNO" "BRSURNAME" "BRADDRESS1" "BRPOSTCODE" "BRDOCTYPE" "BRBATDATE" "BRBATCHNUM"

   Cond
      Case (Equal matchdata nil)
         BRCOMAddMemo "Folder requires manual indexing because auto indexing could not find a match." fabs
         BRComMsg "Folder requires manual indexing because auto indexing could not find a match." "BRCOM.VSS"
"BRCOMSetBRData" 2
         Return 0
      EndCase
      Case (> (Length matchdata) 1)
          DoList (tmp matchdata)
            When (Equal (Nth brdattype_pos tmp) type)
               Push tmp rows
            EndWhen
          EndDoList
```

```
         Assign matchdata (First rows)         ; We are really getting the last row returned by the DB search
because the push reversed the order.
         If matchdata then
            Assign attribs (Send tmplt :TMPLT_BUF2ATTRS matchdata)
            Assign attribs (BRComIdxAttribs attribs type)
            BRComMsg (MakeString "Last row choosen from DB search type " type " : " matchdata) "BRCOM.VSS"
"BRCOMSetBRData" 2
         Else
            BRCOMAddMemo "Folder requires manual indexing because auto indexing could not find a valid match."
fabs
            BRComMsg "Folder requires manual indexing because auto indexing could not find a valid match."
"BRCOM.VSS" "BRCOMSetBRData" 2
            Return 0
         Endif
      EndCase
      Otherwise
         Assign attribs (Send tmplt :TMPLT_BUF2ATTRS (First matchdata))
         Assign attribs (BRComIdxAttribs attribs type)
         SetDocumentInfo fabs :attribs attribs
      EndCase
   EndCond

   ;=====================================================================
   ;Write data to folder attribs. NB if we get this far we have a match
   ;=====================================================================

   Assign batch (GetDocumentInfo fabs :BATCH)

   When attribs

      ;When dealing with BACKFILE DOCUMENTS, stop fields that have been populated from the
      ;backfile control files (*.ctl) been overwritten with empty fields from the database search.

      When (StringEqual batch "BRIMPCCITT")
         DoList (tmp attribs)
            Unless (And (Member (Second tmp) ["" " " nil]                                          :test
'StringEqual)~
                        (Member (First tmp)  ["BRHBISNO" "BRNINO" "BRSURNAME" "BRPOSTCODE" "BRUPRN"] :test
```

```
'StringEqual)~
                        (GetDocumentInfo fabs :field (First tmp)))
               Push tmp attribs2
            EndUnless
         EndDoList
         Assign attribs attribs2
      EndWhen

      SetDocumentInfo fabs :attribs attribs

   EndWhen

   ;=============================================================================
   ;Check all the attribs have been set. NB Backfile docs require different checks.
   ;=============================================================================

   ;NB Backfile folders are routed to archive if they pass the attribute check. All other
   ;folders will be routed to Mail matching for processing. We need to get the backfiles
   ;to directly to archive because the overhead of processing them is too high.

      ;==================
      ;Backfile Documents
      ;==================

   ;; -----------------------------------------------------------------------
   ;; Needs this here for autoindexing...
   ;; Tedy - 29 March 1999
   ;; -----------------------------------------------------------------------
   Assign uprn (GetDocumentInfo fabs :field "BRUPRN")


   ;; -----------------------------------------------------------------------
   ;; Get list of batches that will be forwarded to Archive...
   ;; Tedy - 01 November 1999
   ;; -----------------------------------------------------------------------
   Assign batchIndexToArchive (ReadFromString (GetConfigString ini_file "IndexToArchive" "Batch"))
```

```
   Cond

    ;Case (StringEqual batch "BRSPD")      ;; Tedy - modified again! on 01-Nov-1999 support group of batches.
    Case (Member batch batchIndexToArchive)
      ;BRComMsg "Processing SPD batch..." "BRCOM.VSS" "BRCOMSetBRData" 2
      BRComMsg (MakeString "Processing " batch " batch..." "BRCOM.VSS") "BRCOMSetBRData" 2
      Unless (StringTrim (GetDocumentInfo fabs :field "BRCUSERID") :both " ")
         SetDocumentInfo fabs :attribs [["BRCUSERID" "AUTO"]]
         BRComMsg (MakeString "BRCUSERID set to " batch "AUTO") "BRCOM.VSS" "BRCOMSetBRData" 2
      EndUnless

      If (Member (GetDocumentInfo fabs :field "BRSURNAME") ["" " " nil] :test 'StringEqual)
      Then
         BRComMsg "Folder failed attribute check, SURNAME is not set. Folder sent to Manual Indexing."
"BRCOM.VSS" "BRCOMSetBRData" 2
         BRCOMAddMemo "Folder failed attribute check, SURNAME is not set." fabs
         Return 0
      EndIf

      If (Member uprn ["" " " nil] :test 'StringEqual)
      Then
         BRComMsg "Folder failed the attribute check, UPRN is not set. Sent to Manual Indexing." "BRCOM.VSS"
"BRCOMSetBRData" 2
         BRCOMAddMemo "Folder failed the attribute check, UPRN is not set." fabs
         Return 0
      Else
        ;; Set the BRTEAM as per UPRN range...
        Unless (fboundp 'BRGetGrpSubGrp)
           LoadScript (VSRootAppend "READDATA\\PROJECTS\\BR\\SCRIPT\\BRGETNEW" )
        EndUnless
        Assign newPairList (BRGetGrpSubGrp uprn)
        ;; newPairList format example: [["BRTEAM" "BLUE"]["BRWKGRP" "05"]]      ;; Tedy

        SetDocumentInfo fabs :attribs [["BRTEAM"  (Lookup newPairList "BRTEAM"  2)] ~
                                       ["BRWKGRP" (Lookup newPairList "BRWKGRP" 2)]]
        BRComMsg (MakeString "SPD folder UPRN: " ~
                 (GetDocumentInfo fabs :field "BRUPRN") " assigned to: " ~
                 (GetDocumentInfo fabs :field "BRTEAM") ~
```

```
                " Sub: " (GetDocumentInfo fabs :field "BRWKGRP")) "BRCOM.VSS" "BRCOMSetBRData" 2
      EndIf

      BRComMsg "Folder passed the attribute check. Setting :BRIDXOK to :YES." "BRCOM.VSS" "BRCOMSetBRData" 2
      SetDocumentInfo fabs :BRIDXOK :YES

      ;; ------------------------------------------------------------------------
      ;; Report for iWorld in CSV format...
      ;; Tedy - 15 April 2005
      ;; ------------------------------------------------------------------------
      ;; Load the necessary script files...
      Unless (FBoundP 'BRiWorldCSVReport)
        LoadScript (VSRootAppend "readdata\\projects\\br\\script\\BRDocCSV.fsl")
      EndUnless
      BRiWorldCSVReport fabs
      ;; ... End Report for iWorld in CSV format.
      ;; ------------------------------------------------------------------------

    EndCase

    Case (StringEqual batch "BRIMPCCITT")
      BRComMsg "Checking backfile attributes..." "BRCOM.VSS" "BRCOMSetBRData" 2

      ;Needed for library index BRWFDOCS : BRHBIS BRCTAX BRUPRN BRSURNAME BRPOSTCODE BRTEAM BRCUSERID BRCOID
BRDOCTYPE BRBATCHNUM REQID
      ;Patch to get over problems with missing data in the indexing database.
      Unless (StringTrim (GetDocumentInfo fabs :field "BRTEAM") :both " ")
         SetDocumentInfo fabs :attribs [["BRTEAM" "PRESCAN"]]
         BRComMsg "BRTEAM set to PRESCAN" "BRCOM.VSS" "BRCOMSetBRData" 2
      EndUnless

      Unless (StringTrim (GetDocumentInfo fabs :field "BRCUSERID") :both " ")
         SetDocumentInfo fabs :attribs [["BRCUSERID" "PRESCAN"]]
         BRComMsg "BRCUSERID set to PRESCAN" "BRCOM.VSS" "BRCOMSetBRData" 2
      EndUnless

      ;By this point we are guarenteed the atribs BRHBIS, BRUPRN, BRTEAM, BRCURUSERID, BRDOCTYPE, BRBATCHNUM and
REQID
```

```
      ;If we have BRSURNAME then archive the folder otherwise send it for manual indexings. Its ok to archive the
folder without BRCTAX and BRPOSTCODE.
      ;BRUPRN must have been set because its a mandatory DB field. We check it anyway incase of DB changes.
      If (Member uprn ["" " " nil] :test 'StringEqual)
      Then
         BRComMsg "This Backfile folder has failed the attribute check, UPRN is not set. Folder sent to manual
indexing." "BRCOM.VSS" "BRCOMSetBRData" 2
         BRCOMAddMemo "This Backfile folder has failed the attribute check, UPRN is not set." fabs
         Return 0
      EndIf

      If (Member (GetDocumentInfo fabs :field "BRSURNAME") ["" " " nil] :test 'StringEqual)
      Then
         BRComMsg "This Backfile folder has failed the attribute check, SURNAME is not set. Folder sent to manual
indexing." "BRCOM.VSS" "BRCOMSetBRData" 2
         BRCOMAddMemo "This Backfile folder has failed the attribute check, SURNAME is not set." fabs
         Return 0
      EndIf

      BRComMsg "This Backfile folder has passed the attribute check. Setting :BRIDXOK to :YES." "BRCOM.VSS"
"BRCOMSetBRData" 2
      SetDocumentInfo fabs :BRIDXOK :YES

      ;; No iWorld reort here (backfile is not used any longer!?) - Tedy [April 2005]

    EndCase

    Otherwise
      ;================
      ;Everything else.
      ;================

      ;Check that all attribs have been indexed against the library folder template.
      ;If the folder passes the test then set :BRIDXOK to :YES to route the folder past manual indexing.
      ;If the folder fails then send it to manual indexing and add a memo details explaining why.

      Assign ck_attrs (GetDocumentInfo fabs :attribs)
      Assign ck_tmplt (Send (GetIndexTemplate (GetDocumentInfo fabs :index)) :OBJ_DUP)
```

```
      Assign ck_errs (BRCOMCKData ck_attrs ck_tmplt)

      BRComMsg (MakeString "Folder attribs       " ck_attrs)                              "BRCOM.VSS"
"BRCOMSetBRData" 3
      BRComMsg (MakeString "Folder template      " (Send ck_tmplt :OBJ_GETPROP '@NAME))  "BRCOM.VSS"
"BRCOMSetBRData" 3
      BRComMsg (MakeString "Template Attribs     " (Send ck_tmplt :TMPLT_Q_ATTRS))       "BRCOM.VSS"
"BRCOMSetBRData" 3

      If (Assign ck_data (BRCOMCKData ck_attrs ck_tmplt))
      Then
         BRComMsg (MakeString "This folder failed the Attribute check, sending to manual indexing : " ck_data)
"BRCOM.VSS" "BRCOMSetBRData" 2
         Dolist (tmp ck_data)
            BRCOMAddMemo (StringTrim (MakeString "   " tmp) :both "\(\)") fabs
         EndDoList
         BRCOMAddMemo "This folder has failed the indexing check." fabs
      Else
        ;; ========================================================================
        ;; -> 03-October-2000  Tedy
        ;;    Treat folders by Folder Type, ie. BRFLDTYPE = 'R' (Revenue folders)
        ;;    will be sent to either CTax1 or CTax2 team, depending upon the UPRN
        ;;    division, as defined in brcom.ini
        ;;    Only BRFLDTYPE = 'B' (Benefits folders) will be directed to Housing
        ;;    Benefits Teams (Red, Green, White or Blue)
        ;; ========================================================================

        If (StringEqual (GetDocumentInfo fabs :field "BRFLDTYPE") "R")
           Then
            Assign uprnCTax (GetConfigString ini_file "CTax" "UPRN")
            Unless uprnCTax
               Assign uprnCTax "00033739999999"
               BRComMsg (MakeString "WARNING: No CTax UPRN found in " ini_file ~
                                    " Set to default: 00033739999999")         ~
                                    "BRCOM.VSS" "BRCOMSetBRData" 2
            EndUnless
            If (String> (GetDocumentInfo fabs :field "BRUPRN") uprnCTax)
               Then
```

```
         Assign team "Council Tax 2"
    Else
         Assign team "Council Tax 1"
    EndIf
    Assign wkGrp "0"
    SetDocumentInfo fabs :attribs [["BRTEAM" team] ["BRWKGRP" wkGrp]]
Else
    ;; ---------------------------------------------------------------------------
    ;; Get UPRN division for CTax1 and CTax2... [here for BRFLDTYPE = "B"]
    ;; ---------------------------------------------------------------------------
    ;; Set the BRTEAM as per UPRN range...
    Unless (fboundp 'BRGetGrpSubGrp)
      LoadScript (VSRootAppend "READDATA\\PROJECTS\\BR\\SCRIPT\\BRGETNEW" )
    EndUnless
    Assign newPairList (BRGetGrpSubGrp uprn)
    ;; newPairList format example: [["BRTEAM" "BLUE"]["BRWKGRP" "05"]]

    BRComMsg (MakeString "Folder UPRN newpairList: [" ~
                         (Lookup newPairList "BRTEAM"  2) "  " ~
                         (Lookup newPairList "BRWKGRP" 2) "]") ~
                         "BRCOM.VSS" "BRCOMSetBRData" 2

    Assign team  (Lookup newPairList "BRTEAM"  2)
    Assign wkGrp (Lookup newPairList "BRWKGRP" 2)
    SetDocumentInfo fabs :attribs [["BRTEAM" team] ["BRWKGRP" wkGrp]]
EndIf
;; ===========================================================================

;; -- A low range of UPRNs is not allocated to any of the HB Teams.
;;    This will trap it and dissallow Auto Indexing to take place,
;;    for these cases, thus avoiding folders being sent to Exceptions.
;If (And team wkGrp)
If (And (String> team "") (String> wkGrp ""))
   Then
     BRComMsg (MakeString "Folder type: " ~
              (GetDocumentInfo fabs :field "BRFLDTYPE") " UPRN: " ~
              (GetDocumentInfo fabs :field "BRUPRN") " assigned to: " ~
              (GetDocumentInfo fabs :field "BRTEAM") ~
```

```
                    " Sub: " (GetDocumentInfo fabs :field "BRWKGRP")) "BRCOM.VSS" "BRCOMSetBRData" 2

            BRComMsg "This folder has passed the attribute check. Setting :BRIDXOK to :YES." "BRCOM.VSS"
"BRCOMSetBRData" 2
            SetDocumentInfo fabs :BRIDXOK :YES

            ;; ---------------------------------------------------------------------------
            ;; Report for iWorld in CSV format...
            ;; Tedy - 15 April 2005
            ;; ---------------------------------------------------------------------------
            ;; Load the necessary script files...
            Unless (FBoundP 'BRiWorldCSVReport)
              LoadScript (VSRootAppend "readdata\\projects\\br\\script\\BRDocCSV.fsl")
            EndUnless
            BRiWorldCSVReport fabs
            ;; ... End Report for iWorld in CSV format.
            ;; ---------------------------------------------------------------------------

        Else
            BRComMsg "This folder had no team and workgroup allocation. Sent to Manual Indexing." "BRCOM.VSS"
"BRCOMSetBRData" 2
            ;SetDocumentInfo fabs :BRIDXOK :NO
        EndIf
      EndIf
    EndCase
 EndCond

;  AB1.7 - Start of added code and comments

;  Initialise the Archive Date to empty string, this only necessary because
;  if this attrib is not initialised ViewStar will default it to today's date.
;  This causes confusion as the Archive Date should only be set when the folder gets to the Archive Queue.
   SetDocumentInfo fabs :attribs [["BRARCDATE" ""]]

;  AB1.7 - End of added code and comments

   Return 0
```

```
EndScript



;;============================================================================
;;
;; Check folder attributes against a template.
;;
;; When        Called.
;;
;; Returns     NA
;;
;;============================================================================
DefScript BRCOMCKData

   ;NB tried to use :TMPLT_CK_DATA but it would not work on a process agent. I could only
   ;get it to work in a forward env, so have I have coded a version using documented commands.

   Arguments ck_attrs ck_tmplt

   locals ck_data ck_tmplt_attrs ck_attr ck_mand_attrs attr

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

   Assign ck_tmplt_attrs (Send ck_tmplt :TMPLT_Q_STD_ATTRS)

   ;Check all mandatory attribs exist on the folder.

   DoList (tmp ck_tmplt_attrs)

      ;Mandatory test

      When (Nth 10 tmp)
         If (Assign ck_attr (Lookup ck_attrs (First tmp) nil :test 'StringEqual))
         Then
            When (Member  (Second ck_attr) ["" " " nil])
```

```
              Push (MakeString (First tmp) " Field is mandatory.") ck_data
            EndWhen
         Else
            Push (MakeString (First tmp) " Field is mandatory.") ck_data
         Endif
      EndWhen

   EndDoList

   Return ck_data

EndScript

;=============================================================================
;  BRCOMAddMemo
;  Add memo txt to a folder or document
;
;  WHEN   Script Calls.
;
;  ARGUMENTS:
;  errmsg - an error message.
;
;  OPTIONALS
;  abs -    abs or nil(default)
;           The memo is written to the abs passed. If no abs is passed
;           the current abs open in the fwdenv is used.
;
;  RETURNS: n/a
;
;
;  RULES for ViewStar command GetCurrentViewDoc
;
;     Item in View panel                  Return value if :folder is t            Return value if :folder is nil
;     ------------------------------------------------------------------------------------------------------
;     Nothing in View panel             nil                                       nil
;     Document not in a folder      document                                  document
;     Document in a folder                folder                                        document
;     Folder containing documents       folder                                    folder
```

```
;     Empty folder                      folder                                  folder
;
;=========================================================================
DefScript BRCOMAddMemo                          ;  Add memo txt to a folder or document
Arguments errmsg
Optionals (wp nil)
Locals memo timestamp docabs docmemo fvdoc dvdoc

   ; Convert memo message to a list if isn't already
   Unless (Type errmsg '@list)
     Assign errmsg [errmsg]
   EndUnless

   ; Build up full message text.
   Assign timestamp (BrComDateTimeString (GetCurrentDateTime) "dd/MM/yyyy HH:mm" nil)   ;---BW1 Added---  Need to
pass HH:mm
   Assign (First errmsg) (MakeString timestamp " : " (First errmsg))

   ;If we do not have an abs then try to get one from the current environment.

   Unless wp
      Unless (Assign wp (GetCurrentViewDoc :folder t))         ;This will return a folder, document or nil see
rules above.
         Return nil                                            ;Nothing to add a memo to so return nil
      EndUnless

      If (Isfolder wp) then
         Assign dvdoc (First (SelectFromFolder wp :itemnums [1]))
      Else
         Assign dvdoc wp
      EndIf
   EndUnless

   ; Write the memo to the workpacket.
   Cond
      Case (Type wp '@folder)                                                ;Folder viewdoc
         Assign memo    (GetDocumentInfo wp :MEMO)                           ; (Send x :OBJ_GETPROP :MEMO)
         Assign memo    (Append errmsg memo)
```

```
   Assign docmemo (GetDocumentInfo dvdoc :MEMO)
   Assign docmemo (Append errmsg docmemo)

   SetDocumentInfo wp    :MEMO memo
   SetDocumentInfo dvdoc :MEMO docmemo

   When (StringEqual (MakeString (Send (GetCurrentenvir) :OBJ_GETPROP '@AKO)) "FWDENV")
      UpdateDisplay
   EndWhen
EndCase

Case (Type wp '@document)                                          ;Document viewdoc
   Assign docmemo (GetDocumentInfo dvdoc :MEMO)
   Assign docmemo (Append errmsg docmemo)

   SetDocumentInfo dvdoc  :MEMO docmemo

   When (StringEqual (MakeString (Send (GetCurrentenvir) :OBJ_GETPROP '@AKO)) "FWDENV")
      UpdateDisplay
   EndWhen
EndCase

Case (Type wp '@abstract)                                           ;Abstract
   If (Isfolder wp)                                                ;Folder Abstract
   Then
      WithOpenDocument (fvdoc wp)
         Assign dvdoc    (First (SelectFromFolder fvdoc :itemnums [1]))
         Assign memo     (GetDocumentInfo fvdoc :memo)
         Assign memo     (Append errmsg memo)
         Assign docmemo  (GetDocumentInfo dvdoc :memo)
         Assign docmemo  (Append errmsg docmemo)
         SetDocumentInfo fvdoc :MEMO memo
         SetDocumentInfo dvdoc :MEMO docmemo
      EndWithOpenDocument
   Else
      WithOpenDocument (dvdoc wp)                                   ;Document Abstract
         Assign docmemo (GetDocumentInfo dvdoc :memo)
         Assign docmemo (Append errmsg memo)
```

```
               SetDocumentInfo dvdoc :MEMO docmemo
            EndWithOpenDocument
         Endif
      EndCase

      Otherwise
         Return nil
      EndCase

   EndCond

   Return t

EndScript


;=========================================================================
;
; Sets :CURRENT and :NEXT properties on the current abstract
; and then forwards it according to the value of output
;
;    Arguments
;        current - where the folder is being forwarded from
;        next    - where the folder is being forwarded to
;
;    Optionals
;       output   - specifies the forward action to execute
;       fabs     - folder abstract to be routed           (NB: not necessary if we are routing the current
abstract from a forward environment)
;       reqobj   - request object for folder to be routed (NB: not necessary if we are routing a folder which
already has a route assigned)
;
;
;=========================================================================
DefScript BRComForward
Arguments current next
Use "Route"
Optionals (output :FORWARD) fabs reqobj
```

```
Locals ( env ( GetCurrentEnvir ) )

    Unless fabs
      Assign fabs ( GetCurrentAbstract :folder t )
    EndUnless

;   Set the routing properties :CURRENT and :NEXT value as required

    SetDocumentInfo fabs :CURRENT current
    SetDocumentInfo fabs :NEXT    next

;   Execute the specified forward action (default is ForwardDocument)
    Cond
      Case ( Equal output :ROUTENEW )
        RouteDocument fabs reqobj
      EndCase
      Case ( Equal output :ROUTE )
        RouteDocument fabs
      EndCase
      Case ( Equal output :HOLD )
        HoldDocument
      EndCase
      Case ( Equal output :REJECT )
        RejectDocument
      EndCase
      Otherwise
        ForwardDocument
      EndCase
    EndCond

  Return t

EndScript

;=============================================================================
;
; Called from a script task in the Main Decision Map ( BRDEC.VAM )
; Sets attributes on the folder abstract that always need setting after a forward
```

```
;
;    Arguments
;        fabs    - the folder abstract
;        reqobj  - the folder request object
;
;    Returns t
;============================================================================
DefScript BRComForwardSetAttrs
Arguments fabs reqobj
Locals ( nouserstring  "zzzzzzzzzz" ) currenttime bruserid brluserid brluname brladate brcoid ( env (
GetCurrentenvir ) )
Use "Security"

  Assign currenttime ( GetCurrentDateTime )

; Set the property which records the time at which the user finished processing the folder
  SetDocumentInfo fabs :BRUSER_ENDTIME currenttime

; Last action date
  Assign brladate currenttime

; Set the User ID to the value specified by the property :BRUSERID or to a string which indicates no folder is
not assigned to a specific user
  If ( Assign bruserid ( GetDocumentInfo fabs :BRUSERID ) ) Then
      SetDocumentInfo fabs :BRUSERID nil
  Else
    Assign bruserid nouserstring
  EndIf

; Set up the Last User Id and Last User Name attributes
  Assign brluserid ( First  ( Whoami ) )
  Assign brluname  ( Second ( Whoami ) )

; If the folder has come from Commissioner Map set the BRCOID (Commissioning Officer User ID),
; unless the folder has been completed without going to the Commissioner App for acceptance by a CO.
  When ( StringEqual ( GetDocumentInfo fabs :CURRENT ) :COMMISSIONER )
    If ( StringEqual (Send env :ENV_Q_NAME) "Process Agent") Then
      Assign brcoid ""
```

```
    Else
      Assign brcoid ( First  ( Whoami ) )
    EndIf
  EndWhen

; Set the attribs on the folder
; AB1.4 - Added
  SetDocumentInfo fabs :attribs [ [ "BRUSERID"  bruserid  ] ~
                                  [ "BRLADATE"  brladate  ] ~
                                  [ "BRCOID"    brcoid    ] ]

; Only set the last user info if actioned by a user rather than a Process Agent
  Unless ( StringEqual (Send env :ENV_Q_NAME) "Process Agent")
    SetDocumentInfo fabs :attribs [ [ "BRLUSERID" brluserid ] ~
                                    [ "BRLUNAME"  brluname  ] ]
  Endunless
; AB1.4 - End of Added Code


; Set the REQID attribute if it hasn't already been set, and make sure the workpacket is being Custom Tracked.
; This is only necessary for new folders which are created within an application when using functions
; such as FileNote, AddWordDocument etc. This has to be done here because new folders don't
; have routes (and hence REQID's) until they are forwarded.

  Unless ( GetDocumentInfo fabs :field "REQID" )
    SetDocumentInfo fabs :attribs [ [ "REQID"  (Send reqobj :REQ_Q_ID)  ] ]
  EndUnless

  Return 0

EndScript

;=============================================================================
;
; Called from BRComForwardSetAttrs or a script task in the Mail Matching Map ( BRMATCH.VAM )
; Sets attributes on the folder abstract that always need setting after a forward
;
;    Arguments
```

```
;        fabs    - the folder abstract
;        ignore  - the folder request object
;
;
;    Returns t
;============================================================================
DefScript BRCOMInitWPTracking
Arguments fabs ignore

  Unless ( Send fabs :OBJ_GETPROP :CUST_TRACK )
    InitWorkpacketTracking fabs "BRTRACK"
  EndUnless

  Return t

EndScript

;============================================================================
;  BRCOMSetSearchCriteria
;
;  Description  The purpose of this routine is to set the search criteria
;               on the application so that when a search for a document is
;               performed it happens as indicated by the controls on the
;               Main actionbar.
;
;  WHEN   Script Calls from controls in BRWBCTRL.VSS
;         BRUSER     Combo box
;         BRWBTEAM   Combo box
;         BRWTYPES   Combo box
;         BRSTREAM   Toolbar
;         BRFTYPE    Toolbar
;         BRPRILOW   Combo box
;         BRPRIHI    Combo box
;
;
;  ARGUMENTS:
;  None.
;
```

```
;  OPTIONALS
;  None.
;
;  RETURNS: n/a
;===========================================================================
DefScript BRCOMSetSearchCriteria
Optionals source
Locals abar search_criteria sort_criteria lowpri highpri username worktype  ~  ;SB1.1 Added
         lowest highest pri_status in_target rangeUPRN fromUPRN toUPRN      ~
         ( dlgfile ( VSRootAppend "READDATA\\PROJECTS\\BR\\DIALOGS\\BRWKDESC.CP" ) )  dlg buff wkdesc (env
(GetCurrentEnvir)) ;AB1.8 Added

  Unless ( fboundp 'CSGetVal )
           ;AB1.8 Added
      LoadScript (VSRootAppend "PROJECTS\\COMMON\\CSDATOOL.FSL") :useful ; BW9 Added
  EndUnless
           ;AB1.8 Added

  Assign abar (GetActionBar "Main")
  Assign pri_status (DialogGetItem abar "BRSWPRI")
  Assign in_target  (DialogGetItem abar "BRWBIT")
; Get the stream thats selected

; Get Priority values
  Assign lowpri  (First (Send (Send abar :DIALOG_Q_CTRL "BRLOWPRI" ) :CBOX_Q_ROW  ))
  Assign lowest  (First (Send (Send abar :DIALOG_Q_CTRL "BRLOWPRI" ) :CBOX_Q_ROW 0))
  Assign highpri (First (Send (Send abar :DIALOG_Q_CTRL "BRHIGHPRI") :CBOX_Q_ROW  ))
  Assign highest (First (Send (Send abar :DIALOG_Q_CTRL "BRHIGHPRI") :CBOX_Q_ROW 0))

; Get username value
  Assign username ( First ( Send ( Send abar :DIALOG_Q_CTRL "BRUSER"  ) :CBOX_Q_ROW ) )

; Get Worktype
  Assign worktype ( Second ( Send ( Send abar :DIALOG_Q_CTRL "BRWTYPES"  ) :CBOX_Q_ROW ) )

; AB1.8 - If statement logic replaced by Cond statement
; Build up the search_criteria
```

```
  Cond
    Case (StringEqual worktype "Owned")

;     The user has selected the "Owned" worktype so they want to all work assigned to them regardless
;     of its worktype so put the user id on the search criteria but NOT worktype.
;     Also a different sort criteria is required so assign it to the variable sort_criteria
;     which will be used at the end of this function

      Push [ "BRUSERID"   username ] search_criteria

      If pri_status Then                                                ;CCS11 ADDED
        ; Priority off                                                 ;CCS11 ADDED
        Assign sort_criteria [ ["BRTDATE" "ASC"] ]                      ;CCS11 ADDED
      Else                                                              ;CCS11 ADDED
        ; Prority on                                                   ;CCS11 ADDED
        Assign sort_criteria [ ["BRPRIORITY" "ASC"]["BRTDATE" "ASC"] ]
      EndIf                                                             ;CCS11 ADDED

    EndCase

    Case (StringEqual worktype "Returned LO's")

;     The username can only be a specific user but we also want unassigned stuff
      Push [ "BRUSERID"   ( MakeString "~($='" username "' OR $='zzzzzzzzzz' OR $='')" ) ] search_criteria

      Push [ "BRWKTYPE" worktype ] search_criteria

;     The user has selected the "Returned LO's" worktype so need to check if they want work with a specific
WorkDesc

;     Load the WorkDescription selection Dialog if necessary and save on the environment
      Unless ( Assign dlg ( CSGetVal :WKDESC [[:TARGET env] [:TOPIC :BRDIALOGS]] ) )
        Unless ( Assign dlg ( ObjLoad dlgfile ) )
          BRComMsg "Failed to locate Referral Dialog." "BRREFER.VSS" "BRRefer" 1
          Return nil
        EndUnless
       CSPutVal :WKDESC dlg [[:TARGET env] [:TOPIC :BRDIALOGS]]
      EndUnless
```

```
      If dlg Then
        Assign buff (BRComGetDocTypes "Returned LO's")
        SetBuffer dlg (Push ["All Work Descriptions"] buff) "wkdescdl"
        Send (GetDatalist dlg "wkdescdl") :DL_S_PICK 0
        Assign wkdesc (BRCOMPopup dlg)
      Else
        Assign wkdesc "All Work Descriptions"
      EndIf

;     If the user has selected a specific work description add it to the search criteria

      Unless (StringEqual wkdesc "All Work Descriptions")
       Push [ "BRWKDESC"   wkdesc ] search_criteria
      EndUnless

    EndCase

    Otherwise

;     Must be a specific worktype
      Push [ "BRWKTYPE" worktype ] search_criteria
    EndCase

  EndCond

; AB1.8 - End of modified section


  Unless  pri_status                                                              ;CCS11 ADDED

  ; Priority on                                                                   ;CCS11 ADDED

  ; Now the priority
  ; Can be over complete range, a single value or a subrange

  Cond
```

```
    Case ( And ( Equal highpri highest ) ( Equal lowpri lowest ) )
      Value nil
    EndCase

    Case ( Equal highpri lowpri )
      Push [ "BRPRIORITY" lowpri ] search_criteria
    EndCase

    Case (And (Equal highpri highest) (Not (Equal lowpri lowest)))
      Push [ "BRPRIORITY" (MakeString "~($<='" lowpri"')") ] search_criteria
    EndCase

    Case (And (Equal lowpri lowest) (Not (Equal highpri highest)))
      Push [ "BRPRIORITY" (MakeString "~($>='" highpri "')")] search_criteria
    EndCase

    Otherwise
      Push [ "BRPRIORITY" ( MakeString "~($>='" highpri "' & $<='" lowpri "')" ) ] search_criteria
    EndCase

  EndCond

  EndUnless                                                                                    ;CCS11 ADDED

  ; If the in target check box is check in workbench then restrict the folders pulled back to those that can
still be   ;CCS11 ADDED
  ; completed within there target date(This excludes out of date work!).
      ;CCS11 ADDED

;  When in_target
       ;CCS11 ADDED
;     Push ["BRTDATE" (MakeString "~($>='" (FormatDateTime (GetCurrentDateTime) "MM/dd/yyyy" NIL) "')")]
search_criteria ;CCS11 ADDED
;  EndWhen
       ;CCS11 ADDED

;; ---------------------------------------------------------------------------
;; When in "Assigned mode" add the UPRN based criteria here, or replace the
```

```
;; existing UPRN based criteria, as this could be a new user.
;; When in "Opened mode" remove the UPRN based criteria and leave the rest
;; ie. WorkTypw and WFSTATUS unchanged...
;; Tedy - March 1999
;; ------------------------------------------------------------------------

   Unless (Equal source :MAINBAR)
     ;; Get the UPRN range for this user id...
     Assign rangeUPRN (BRSetUserSearchCriteria username)
     Assign fromUPRN (First  rangeUPRN)
     Assign toUPRN   (Second rangeUPRN)
     ;; If user is assigned UPRN range add it to the search_criteria, so that
     ;; the Next button works on that range exclusively...
     If (And fromUPRN toUPRN)
       Then
         Assign search_criteria ~
                (Push [ "BRUPRN" (MakeString "~$['" ~
                                  fromUPRN "','" ~
                                  toUPRN   "']")] search_criteria)
     EndIf
   EndUnless
;; ------------------------------------------------------------------------

; Set the search_criteria
  SetProcessOption :search_criteria search_criteria

; Set the sort_criteria, if the sort_criteria variable has no value assign the default
  If sort_criteria
    Then
      SetProcessOption :sort_criteria  sort_criteria
  Else

      If pri_status Then
     ;CCS11 ADDED
        ; Priority off
     ;CCS11 ADDED
        SetProcessOption :sort_criteria  [["BRWKTYPE" "ASC"] ["BRTDATE" "ASC"] ["BRUSERID" "ASC"]]
     ;CCS11 ADDED
```

```
      Else
     ;CCS11 ADDED
        SetProcessOption :sort_criteria  [["BRWKTYPE" "ASC"] ["BRPRIORITY" "ASC"]["BRTDATE" "ASC"] ["BRUSERID"
"ASC"]]
      EndIf
     ;CCS11 ADDED

  EndIf

  Return

EndScript ;BRCOMSetSearchCriteria


;=============================================================================
;  BRCOMSetupAppCustomData
;
;  Description:  Set up all application custom data on the environment.
;
;  WHEN   Script Calls from controls in BRWBCTRL.VSS
;
;  ARGUMENTS:
;  None.
;
;  OPTIONALS
;  None.
;
;  RETURNS: n/a
;=============================================================================
DefScript BRCOMSetupAppCustomData
MoreArguments ignore
Locals (env (GetCurrentEnvir)) ~
       (ini_file (VSRootAppend "VS_BIN\\BRCOM.INI")) ~
       usersList (userID (First(WhoAmI))) queuelist

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

```
   Unless (And (fboundp 'CSPutVal) (fboundp 'CSGetVal))
      LoadScript (VSRootAppend "PROJECTS\\COMMON\\CSDATOOL.FSL") :mandatory ; BW9 Added
   EndUnless

   ; Set Custom data.

   ; NB the following script function sets up these properties.
   ;      :BRCURRUSERGROUP  - the group the current user belongs to
   ;      :BRUSERGROUPS     - usergroups that share the same profile as the current user
   ;      :BRGROUPMEMBERS   - usergroups that share the same profile as the current user and the users within the
groups
   ;      :BRWBUSERGROUPS   - usergroups that share the "RBOFF" profile
   ;      :BRWBGROUPMEMBERS - usergroups that share the "RBOFF" profile and the users within the groups
   ;      :BRCOUSERGROUPS   - usergroups that share the "COMMOFF" profile
   ;      :BRCOGROUPMEMBERS - usergroups that share the "COMMOFF" profile and the users within the groups
   Unless (And ~
               (CsGetVal :BRCURRUSERGROUP   [[:TARGET env] [:TOPIC :CONFIG_VALS]])~
               (CsGetVal :BRUSERGROUPS      [[:TARGET env] [:TOPIC :CONFIG_VALS]])~
               (CsGetVal :BRGROUPMEMBERS    [[:TARGET env] [:TOPIC :CONFIG_VALS]])~
               (CsGetVal :BRWBUSERGROUPS    [[:TARGET env] [:TOPIC :CONFIG_VALS]])~
               (CsGetVal :BRWBGROUPMEMBERS  [[:TARGET env] [:TOPIC :CONFIG_VALS]])~
               (CsGetVal :BRCOUSERGROUPS    [[:TARGET env] [:TOPIC :CONFIG_VALS]])~
               (CsGetVal :BRCOGROUPMEMBERS  [[:TARGET env] [:TOPIC :CONFIG_VALS]]))
      BRComSetGroupUsers
   EndUnless

;; ---------------------------------------------------------------------------
;; Add a default of Process Mode to "Assigned" (by UPRN range)
;; Users have the option of changing this later in the session,
;; through a Select Process... function (in the Workbench application).
;; Users can also control the WorkType and WFSTATUS. These are added to
;; the search criteria in "Find", such that users can be more specific in
;; the way they pickup work from the queues.
;; This new system replaces the current search by given REQID list.
;;
;; Tedy - March 1999
;; ---------------------------------------------------------------------------
```

```
;Inspect["About to run BRSetUserSearchCriteria userID - Init..." userID]
   BRSetUserSearchCriteria userID "Init"

;; ------------------------------------------------------------------------------
;; Rather than allowing team Leaders and Deputies only, access the queues count
;; function, use brcom.ini file to set a list of users IDs that are allowed access...
;; Also note that the "Count Folders..." option may be disabled, system wise if a
;; file named BRNOCNT.TXT is placed in the VS_BIN directory.
;; Tedy - March 1999
;; ------------------------------------------------------------------------------

   Assign usersList (ReadFromString (GetConfigString ini_file "CountFolders" "Users"))
   If (Member userID usersList)
     Then
       CSPutVal :BRCOUNTOK [[:ASSIGNED t]   [:ALL t  ]] ~
                           [[:TARGET env]   [:TOPIC :CONFIG_VALS]]
   Else
       CSPutVal :BRCOUNTOK [[:ASSIGNED nil] [:ALL nil]] ~
                           [[:TARGET env  ] [:TOPIC :CONFIG_VALS]]
   EndIf

;; ------------------------------------------------------------------------------

   ;   :BRWORKTYPES     - worktypes set up in custom data.

   Unless (CsGetVal :BRWORKTYPES [[:TARGET env] [:TOPIC :CONFIG_VALS]])
      CSPutVal :BRWORKTYPES (BRSQLExecSP "sp_brselworktypes" nil) [[:TARGET env] [:TOPIC :CONFIG_VALS]]
   EndUnless

   ;   :BRWORKTYPES     - worktypes set up in custom data without the first column (i.e. the 'ID' column).

   Unless (CsGetVal :BRWORKTYPES2 [[:TARGET env] [:TOPIC :CONFIG_VALS]])
      CSPutVal :BRWORKTYPES2 ( @sort_list (BRSQLExecSP "sp_brselwtnotfirst" nil) 'string< '@First ) [[:TARGET
env] [:TOPIC :CONFIG_VALS]]
   EndUnless

   ;   :BRPRIORITIES    - Folder priorties set up in custom data.
```

```
   Unless (CsGetVal :BRPRIORITIES [[:TARGET env] [:TOPIC :CONFIG_VALS]])
      CSPutVal :BRPRIORITIES (BRSQLExecSP "sp_brselpriority" nil) [[:TARGET env] [:TOPIC :CONFIG_VALS]]
   EndUnless

   ;   :BRFLDTYPES      - Folder types set up in custom data.

   Unless (CsGetVal :BRFLDTYPES [[:TARGET env] [:TOPIC :CONFIG_VALS]])
      CSPutVal :BRFLDTYPES (BRSQLExecSP "sp_brselfoltypes" nil) [[:TARGET env] [:TOPIC :CONFIG_VALS]]
   EndUnless

   ;   :BRSTREAMS      - Folder types set up in custom data.

   Unless (CsGetVal :BRSTREAMS [[:TARGET env] [:TOPIC :CONFIG_VALS]])
      CSPutVal :BRSTREAMS (BRSQLExecSP "sp_brselstreams" nil) [[:TARGET env] [:TOPIC :CONFIG_VALS]]
   EndUnless

;; ---------------------------------------------------------------------------------
;; The following was moved here from brfind.vss BRFind script.
;; Tedy 24-March-1999
;; Set custom exclusions for "Find" on BRTEAM and STATUS fields...
; Get the list of queues that users are allowed to open folders from

   Unless (Fboundp 'BRGetSelectiveQFind)
     LoadScript (VSRootAppend "readdata\\projects\\br\\script\\brfind")
   EndUnless
   Assign queuelist (BRGetSelectiveQFind :STARTUP)
;; ---------------------------------------------------------------------------------

   Return t

EndScript ;BRCOMSetupAppCustomData


;=========================================================================
;
; This Function is called to set up information about users and groups which belong
; to the same profile as the current user and also the following profiles
;
```

```
;      "RBOFF"   - Revenues and Benefits Officers
;      "COMMOFF" - Commissioning Officers
;
; The following properties are set up as custom data on the environment
;
;      :BRCURRUSERGROUP  - the group the current user belongs to
;      :BRUSERGROUPS     - usergroups that share the same profile as the current user
;      :BRGROUPMEMBERS   - usergroups that share the same profile as the current user and the users within the
groups
;
;      :BRWBUSERGROUPS   - usergroups that share the "RBOFF" profile
;      :BRWBGROUPMEMBERS - usergroups that share the "RBOFF" profile and the users within the groups
;      :BRCOUSERGROUPS   - usergroups that share the "COMMOFF" profile
;      :BRCOGROUPMEMBERS - usergroups that share the "COMMOFF" profile and the users within the groups
;
; Arguments - None
;
;=========================================================================
DefScript BRComSetGroupUsers

  MoreArguments ignore

  Locals ( env ( GetCurrentEnvir ) ) ret wbusergroups wbgroupmembers ;; M.L. 1.6 added two variables wbusergroups
wbgroupmembers

;  Set up the custom data for the Revenues and Benefits Officers profile

   Assign ret ( BRComGetGroupUsers "RBOFF" )

   CSPutVal :BRWBUSERGROUPS    ( First ret  )  [[:TARGET env] [:TOPIC :CONFIG_VALS]]
   CSPutVal :BRWBGROUPMEMBERS  ( Second ret )  [[:TARGET env] [:TOPIC :CONFIG_VALS]]
   When ( Third ret )
     CSPutVal :BRCURRUSERGROUP ( Third ret  )  [[:TARGET env] [:TOPIC :CONFIG_VALS]]
     CSPutVal :BRUSERGROUPS    ( First ret  )  [[:TARGET env] [:TOPIC :CONFIG_VALS]]
     CSPutVal :BRGROUPMEMBERS  ( Second ret )  [[:TARGET env] [:TOPIC :CONFIG_VALS]]
   EndWhen

;; *************************************************************************
```

```
;; M.L. 1.6 Start. Get the usergroup and the members in that group for the R & B officers
;; in order to combine them with the commissioners group.
;; NOTE: The Commissioner attributes used to be set first but they are now second as
;; we are configuring two of the Commissioner releted data items from the
;; workbench releated attributes.
;; ********************************************************************************

   Assign wbusergroups (First ret)
   Assign wbgroupmembers (Second ret)

;; ************
;; M.L. 1.6 End
;; ************

;  Set up the custom data for the Commissioning Officers profile

   Assign ret ( BRComGetGroupUsers "COMMOFF" )

   CSPutVal :BRCOUSERGROUPS    ( First ret  )  [[:TARGET env] [:TOPIC :CONFIG_VALS]]
   CSPutVal :BRCOGROUPMEMBERS  ( Second ret )  [[:TARGET env] [:TOPIC :CONFIG_VALS]]

   When ( Third ret )
     CSPutVal :BRCURRUSERGROUP ( Third ret  )  [[:TARGET env] [:TOPIC :CONFIG_VALS]]

;; ********************************************************************************
;; M.L. 1.6 Start. Comment out the old first and second lists which returned just
;; the commissioners group. Added wbusergroup and wbgroupmembers to the commissioners
;; group
;; ********************************************************************************

;;     CSPutVal :BRUSERGROUPS    ( First ret  )  [[:TARGET env] [:TOPIC :CONFIG_VALS]]
;;     CSPutVal :BRGROUPMEMBERS  ( Second ret )  [[:TARGET env] [:TOPIC :CONFIG_VALS]]

     CSPutVal :BRUSERGROUPS    (Append ( First ret  ) wbusergroups) [[:TARGET env] [:TOPIC :CONFIG_VALS]]
     CSPutVal :BRGROUPMEMBERS  (Append ( Second ret ) wbgroupmembers)  [[:TARGET env] [:TOPIC :CONFIG_VALS]]

;; ************
;; M.L. 1.6 End
```

```
;; ************

      EndWhen

   Return t

EndScript ;BRComSetGroupUsers


;============================================================================
;
; This Function is called to get user and group info about a profile
;
; Arguments - profile
;
; Returns a list of three items
;
;      usergroups     - usergroups that share the same profile as the current user
;      groupmembers   - user groups and the users within the groups
;      currusergroup  - the group the current user belongs to
;
;
;============================================================================
DefScript BRComGetGroupUsers

  Arguments profile

  Use "Security"

  Locals  curruser app usergroup usergroups groupmembers users usernames temp allusers currusergroup

; Load neccessary script files

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

  Unless ( fboundp 'CSPutVal )
```

```
;     LoadScript (VSRootAppend "PROJECTS\\COMMON\\CSDATOOL.VSS") :mandatory ; BW9 Commented Out
      LoadScript (VSRootAppend "PROJECTS\\COMMON\\CSDATOOL.FSL") :mandatory ; BW9 Added
  EndUnless

; Get the Current user's ID

  Assign curruser ( First ( WhoAmI ) )

; Call stored procedure with the profile name as an argument,
; returns the list of groups assigned to the profile

  Assign usergroups ( BRSQLExecSP "sp_brselgroups" [profile] )

  Assign allusers ( QueryAllUsers )

  DoList ( usergroup usergroups )

    When ( Assign users ( GetGroupMembers ( First usergroup ) ) )

      Assign usernames nil

      DoList ( user users )
        Push [ ( First user ) ( Lookup allusers ( First user ) 2 ) ] usernames
      EndDoList

      Push [ ( First usergroup ) ( @sort_list usernames 'string< '@Second ) ] groupmembers

      When ( Lookup users curruser 1 )
        Assign currusergroup usergroup
      EndWhen

     EndWhen

  EndDoList

  Return [ usergroups groupmembers currusergroup]

EndScript
```

```
;;=============================================================================
;;
;; Process script messages in different environments
;;
;;
;; When                 adhoc from script functions
;;
;;
;; Arguments            msg      Error Message.
;;
;; Optionals            file     Script file called from.
;;                      func     Script function with the error.
;;                      level    Error level.
;;
;; Returns              t or nil
;;
;;Code Concerns         None
;;
;;UnitTest Function     TRComMsg
;;
;;Unit Test File        <ReadRoot>\readdata\projects\br\unittest.vss
;;
;;=============================================================================
DefScript BRComMsg                  ;Deal with error messages.

   Arguments msg

   Optionals file func level

   ; Find out what environment we are running in and display message accordingly.

   If (StringEqual (MakeString (Send (GetCurrentenvir) :OBJ_GETPROP '@AKO)) "FWDENV")
   Then
      ShowMessage (MakeString msg)
   Else
      Send '@wfmgr_create :WFM_SERVER_MSG (MakeString msg) level
   Endif
```

```
   Return t

EndScript

;=============================================================================
;
;  Function that takes a passed string and returns a string padded with
;  blanks to a specified length.  Blanks are appended to the end (default)
;  or start of the string according to the keyword specified.
;
;=============================================================================
DefScript BRComPadString

;  Function that takes a passed string and returns a string padded with
;  blanks to a specified length.  Blanks are appended to the end (default)
;  or start of the string according to the keyword specified.

   Arguments txt len

   Optionals (keywd :END)

   Locals blanks blanks80 preblanks postblanks midpos txtlen

;  If text is already as long as (or longer) than the required length, return
;  it as it is

   Assign txtlen (Length txt)
   Unless (< txtlen len)
      Return txt
   EndUnless

;  String is shorter than required.  Work out how many blanks we need.

;  Initialise a string with 80 blanks.  N.B. we have to use this instead
;  of using NewString since NewString causes an error "Unmatched Keyword
;  :initial-element" when we try to create a blank string
```

```
   Assign blanks80 "                                                                                "
   Assign blanks (SubString blanks80 0 (- len txtlen))

   Switch keywd
      Case :START
         Return (MakeString blanks txt)
      EndCase
      Case :END
         Return (MakeString txt blanks)
      EndCase
      Case :CENTRE
         Assign midpos (Round (/ (Length blanks) 2))
         Assign preblanks (SubString blanks 0 midpos)
         Assign postblanks (Substring blanks midpos (+ (Length blanks) 1))
         Return (MakeString preblanks txt postblanks)
      EndCase
      Otherwise
         Return nil
      EndCase
   EndSwitch
EndScript

;=============================================================================
;
; ccs: BW1
;
; If the datetime argument is already a string the string is returned
; but If the datetime argument is in internal date format it is converted
; to the format specified by the format string
;
; Required because VS is inconsistent about the data type
; it uses to store datetime attributes, they can be strings or internal format
; ( VS converts to strings when the SaveAttribs function is used in low level code
;   when copying attribs from a data panel back to the folder )
;
; Arguments datetime - datetime that requires formatting
;           Optionals format   - format string to use when formatting the date
;                      (N.B only used if datetime argument is in VS internal date format)
```

```
;           Optionals timezone - timezone
;                     (N.B only used if datetime argument is in VS internal date format)
;
;           Description: Convert Date and Time into a string which displays dd/MM/yyyy HH:mm:ss
;
;=====================================================================
DefScript BRComDateTimeString

  Arguments datetime
  Optionals (format "dd/MM/yyyy HH:mm:ss") (timezone nil)

 ; Only one argument needs to be passed.
 ; Set the format to be dd/MM/yyyy HH:mm:ss, if different then pass
 ; in the required format i.e. dd/MM/yyyy HH:mm in the function call.
 ; Pass in the required argument formats

  Cond
    Case ( Type datetime '@string ) ; Pass in a string could be dd/mm/yy
      Return (FormatDateTime (ParseDateTime datetime format timezone) format timezone)  ; if a string return
dd/mm/yyyy
    EndCase
    Case ( Type datetime '@cons )                                                       ; Else in internal
format, pass in format dd/MM/yyyy and display dd/MM/yyyy
      Return ( FormatDateTime datetime format timezone )
    EndCase

  EndCond

EndScript

;=====================================================================
;
; ccs: BW1
;
; Arguments datetime - datetime that requires formatting
;           format   - format string to use when formatting the date
;                      (N.B only used if datetime argument is in VS internal date format)
;           timezone - timezone
```

```
;                     (N.B only used if datetime argument is in VS internal date format)
;
;           Description: Convert Date and Time into Viewstar internal format dd/MM/yyyy HH:mm:ss
;
;=============================================================================
DefScript BRComInternalDateTime

  Arguments datetime
  Optionals (format "dd/MM/yyyy HH:mm:ss") (timezone nil)

  Cond
    Case ( Type datetime '@string )
      Return ( ParseDateTime datetime format timezone )                              ; if string put into
internal format
    EndCase
    Case ( Type datetime '@cons )
       Return (ParseDateTime (FormatDateTime datetime format timezone) format timezone) ; if in internal format
format it to dd/MM/yyyy HH:mm:ss and parse into internal format
    EndCase

  EndCond

EndScript

;=============================================================================
;
; ccs: BW1
;
; Arguments datetime - datetime that requires formatting
;           format   - format string to use when formatting the date
;                      (N.B only used if datetime argument is in VS internal date format)
;           timezone - timezone
;                      (N.B only used if datetime argument is in VS internal date format)
;
;           Description: Convert Date Only into a string which displays dd/MM/yyyy
;
;=============================================================================
DefScript BRComDateString
```

```
  Arguments datetime
  Optionals (format "dd/MM/yyyy") (timezone nil)

  Cond
    Case ( Type datetime '@string )
      Return (FormatDateTime (ParseDateTime datetime format timezone) format timezone)
    EndCase
    Case ( Type datetime '@cons )
      Return ( FormatDateTime datetime format timezone )
    EndCase

  EndCond

EndScript

;=====================================================================
;
; ccs: BW1
;
; Arguments datetime - datetime that requires formatting
;           format   - format string to use when formatting the date
;                      (N.B only used if datetime argument is in VS internal date format)
;           timezone - timezone
;                      (N.B only used if datetime argument is in VS internal date format)
;
;           Description: Convert Date Only into Viewstar internal format dd/MM/yyyy
;
;=====================================================================
DefScript BRComInternalDate

  Arguments datetime
  Optionals (format "dd/MM/yyyy") (timezone nil)

  Cond
    Case ( Type datetime '@string )
      Return ( ParseDateTime datetime format timezone )
    EndCase
```

```
    Case ( Type datetime '@cons )
      Return (ParseDateTime (FormatDateTime datetime format timezone) format timezone)
    EndCase

  EndCond

EndScript

;=============================================================================
;
; Copies documents from a view only application into the current folder in a
; process layout application
;
; This function stubs the function attached to the standard VS Folder -> AddCopy
; menu option (i.e fldr_addcopy which is undocumented)
;
; Also uses function GetDocumentsFromView which is the menu precheck function for
; Folder -> AddCopy to check there are documents selected for copy
;
; Arguments - None
;
; Returns   - t or nil
;
;=============================================================================
DefScript BRComAddCopy

  MoreArguments ignore

  Locals fvdoc createflag

; Display the hourglass cursor

  ShowBusyCursor

; Check if the user has selected documents to be copied

  Unless ( GetDocumentsFromView t )
    BRComMsg "No documents selected" "BRCOM.VSS" "BRComAddCopy" 2
```

```
    Return nil
  EndUnless

; Check if there is a folder open

  Unless ( Assign fvdoc ( GetCurrentViewDoc :folder t ) )

;   Can't create new folders in Commissioner so just display an error message

;   When ( Equal ( Send ( GetCurrentEnvir ) :ENV_Q_NAME ) "Commissioner" )   ;BW15 Commented out.
    When ( Equal ( Send ( GetCurrentEnvir ) :ENV_Q_NAME ) "Client Officer" ) ;BW15 Added
      BRComMsg "No folder currently open" "BRCOM.VSS" "BRComAddCopy" 2
      Return nil
    EndWhen

    Unless ( Assign createflag ( BRComCreateNewFolder ) )
      BRComMsg "Could not create a New Folder" "BRCOM.VSS" "BRComAddCopy" 2
      Return nil
    EndUnless

  EndUnless

; Copy documents from ViewOnly app into process layout folder

  fldr_addcopy

  UpdateAbstract (GetCurrentViewDOC :folder t)

; When createflag
                              ; BW6 Commented Out
;   BRCOMShowmessagebox :title "New Folder" :text ["You should 'Map From' LGVision to complete New Folder setup"]
:BUTTONS :OK :ICON :EXCLAIM  ; BW6 Commented Out
; EndWhen
                              ; BW6 Commented Out

  Return t

EndScript
```

```
;=========================================================================
;
; Called by functions which add new documents to a folder,
; checks if there is a folder open if not asks the user if they want to create
; a new one.
;
; This function stubs the function attached to the standard VS Folder -> New
; menu option (i.e fldr_new which is undocumented);
;
; Arguments - None
;
; Returns   - t or nil
;
;=========================================================================
DefScript BRComCreateNewFolder

  MoreArguments ignore

  Locals fvdoc msgtext1 msgtext2 createflag dlg ( env (GetCurrentEnvir ) ) brteam brwktype brstream brweight
brpriority brtdate teamcbox wktypecbox data wktypedata ~
         ( dlgfile ( VSRootAppend "READDATA\\PROJECTS\\BR\\DIALOGS\\BRNWFLDR.CP" ) ) (nouserstring "zzzzzzzzzz")

; Display the hourglass cursor

  ShowBusyCursor

; Make sure necessary scripts are loaded

 Unless (fboundp 'BRTrackOpenTimes)                                          ;  AB1.4 - Added
   LoadScript (VSRootAppend "readdata\\projects\\br\\script\\brtrack.vss") :useful    ;  AB1.4 - Added
 EndUnless                                                                   ;  AB1.4 - Added

; Check if there is a folder open, if not create a new one

  Unless ( Assign fvdoc ( GetCurrentViewDoc :folder t ) )

;   Ask the user if they want to create a new folder
```

```
    Assign msgtext1 ( MakeString "There is no folder currently open. " )
    Assign msgtext2 ( MakeString "Create a New Folder ?" )
    Unless ( Equal ( BRCOMShowMessageBox :title "Create New Folder" :text [msgtext1  msgtext2 ] :BUTTONS :YESNO
:ICON :QUEST )  :YES )
      Return nil
    EndUnless

;   Make the user select a team and worktype for the new folder
;   Load the Team/Worktype Dialog if necessary and save on the on the environment

    Unless ( Assign dlg ( CSGetVal :NEW_FOLDER [[:TARGET env] [:TOPIC :BRDIALOGS]] ) )
      Unless ( Assign dlg ( ObjLoad dlgfile ) )
        Return nil
      EndUnless

      Assign teamcbox    ( Send dlg :DIALOG_Q_CTRL "Team" )
      Assign wktypecbox  ( Send dlg :DIALOG_Q_CTRL "Worktype" )
      Send teamcbox   :CBOX_S_BUFFER ( CSGetVal :BRWBUSERGROUPS [[:TARGET env] [:TOPIC :CONFIG_VALS]] )
      Send teamcbox   :CBOX_S_PICK 0
      Send wktypecbox :CBOX_S_BUFFER ( CSGetVal :BRWORKTYPES2   [[:TARGET env] [:TOPIC :CONFIG_VALS]] )
;      Send wktypecbox :CBOX_S_PICK 0
      Send wktypecbox :CBOX_S_PICK nil                          ; Changed to NIL AndyB 10/06/1998 (HelpDesk Call
1190)

      CSPutVal :NEW_FOLDER dlg [[:TARGET env] [:TOPIC :BRDIALOGS]]
    EndUnless

    Assign wktypecbox  ( Send dlg :DIALOG_Q_CTRL "Worktype" )    ; Added AndyB  10/06/1998 (HelpDesk Call 1190)
    Send wktypecbox :CBOX_S_PICK nil                             ; Added AndyB  10/06/1998 (HelpDesk Call 1190)

    Unless ( Assign data ( BRCOMPopup dlg ) )
      Return nil
    EndUnless

    Assign brteam     ( First  data )
    Assign brwktype   ( Second data )
```

```
    Unless ( Assign wktypedata ( Lookup ( CSGetVal :BRWORKTYPES2 [[:TARGET env] [:TOPIC :CONFIG_VALS]] ) brwktype
) )
      Return nil
    EndUnless

    Assign brstream   ( Nth 2 wktypedata )
    Assign brweight   ( Nth 3 wktypedata )
    Assign brpriority ( Nth 4 wktypedata )
;   Assign brtdate    ( DateTime+ ( GetCurrentDateTime ) ( Nth 5 wktypedata ) :day )   ;AB1.1 Commented Out
    Assign brtdate    ( BRCOMTargetDate ( GetCurrentDateTime ) ( Nth 5 wktypedata ) )  ;AB1.1 Added

;   fldr_new is an undocumented VS function, it is the function called when the New option is selected from the
Folder Menu (a Standard VS feature)

    fldr_new
    Assign fvdoc ( GetCurrentViewDoc :folder t )

    BRTrackOpenTimes ( GetCurrentAbstract :folder t )      ;  AB1.4 - Added

;   Set the folder's attribs

    SetDocumentInfo fvdoc :attribs [ [ "BRWKDESC"   "New Folder"         ] ~
                                     [ "BRSTATUS"   "WORKBENCH"          ] ~
                                     [ "BRBATDATE"  (GetCurrentDateTime) ] ~
                                     [ "BRBATCHNUM" 0                    ] ~
                                     [ "BRUSERID"   nouserstring         ] ~
                                     [ "BRTEAM"     brteam               ] ~
                                     [ "BRWKTYPE"   brwktype             ] ~
                                     [ "BRSTREAM"   brstream             ] ~
                                     [ "BRWEIGHT"   brweight             ] ~
                                     [ "BRPRIORITY" brpriority           ] ~
                                     [ "BRFLDTYPE"  (Nth 10 wktypedata)  ] ~
                                     [ "BRTDATE"    brtdate              ]]

  EndUnless

  Return t
```

```
EndScript

;============================================================================
;
;  Called by OK button of new folder dialog
;
;============================================================================
DefScript BRComNewFolderOK

  Arguments ignore dlg ignore

  ReturnFromPopup [ (Second ( Send ( Send dlg :DIALOG_Q_CTRL "Team" )     :CBOX_Q_ROW ) ) ~
                    (First  ( Send ( Send dlg :DIALOG_Q_CTRL "Worktype" ) :CBOX_Q_ROW ) ) ]

EndScript
;============================================================================
;
;  Called when the script is called from the WorkBench Application
;
;============================================================================
DefScript BRCOMHistMenu

  MoreArguments ignore

  Unless (GetCurrentAbstract :folder t)
     BRComMsg "No folder currently selected." "BRCOM.VSS" "BRHistMenu" 2
    Return
  EndUnless

  Unless (Send (GetRoute (GetCurrentAbstract :folder t)) :REQ_Q_ID)
     BRComMsg "Cannot display history for new folders." "BRCOM.VSS" "BRHistMenu" 2
    Return
  EndUnless

  BRCOMHistDisplayPopup nil

EndScript
```

```
;============================================================================
;
;  Called when the user selects the Status Button from the View Folder History
;  dialog
;
;
;============================================================================

DefScript BRCOMHistStatus

  Arguments ignore dlg ignore

  Locals buffer (reqid (Send (GetRoute (GetCurrentAbstract :folder t)) :REQ_Q_ID))

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

  Assign buffer (BRSQLExecSP "SP_BRHistSelStatus" [reqid])

  BRCOMHistDisplayRows dlg buffer :REQUEST

  Return nil
EndScript

;============================================================================
;
;  Called when the user selects the History Button from the View Folder History
;  dialog
;
;============================================================================

DefScript BRCOMHistHist

  Arguments ignore dlg ignore

  Locals buffer (reqid (Send (GetRoute (GetCurrentAbstract :folder t)) :REQ_Q_ID))
```

```
  Unless (FBoundP 'BRSQLExecSP)
    LoadScript (VSRootAppend "readdata\\projects\\br\\script\\brsql.vss") :mandatory
  EndUnless

  Assign buffer (BRSQLExecSP "SP_BRHistSelHist" [reqid])

  BRCOMHistDisplayRows dlg buffer :REQHIST

  Return nil

EndScript

;=============================================================================
;
;  Called when the user selects the Users Button from the View Folder History
;  dialog
;
;=============================================================================

DefScript BRCOMHistUsers

  Arguments ignore dlg ignore

  Locals buffer (reqid (Send (GetRoute (GetCurrentAbstract :folder t)) :REQ_Q_ID))

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

  Assign buffer (BRSQLExecSP "SP_BRHistSelUsers" [reqid])

  BRCOMHistDisplayRows dlg buffer :CUSTOM "BRUSER"

  Return nil
EndScript

;=============================================================================
;
```

```
;  Called from BRCOMHistMenu inorder to popup the dialog box in the Workbench
;  application Window.
;
;============================================================================
DefScript BRCOMHistDisplayPopup

  MoreArguments ignore

  Locals dlg (dlgfile (VSRootAppend "readdata\\projects\\br\\dialogs\\brcomhst.cp")) tmplt buffer


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

 Unless ( Assign dlg (CSGetVal :BRHISTSEARCH [[:TARGET (GetCurrentEnvir)][:TOPIC :BRDIALOGS]]))

    Unless (Assign dlg (ObjLoad dlgfile))
      BRComMsg "Can not load the indexing object file." "BRCOM.VSS" "BRCOMHistDisplayDlg" 2
      Return nil
    EndUnless

    CSPutVal :BRHISTSEARCH dlg [[:TARGET (GetCurrentEnvir)][:TOPIC :BRDIALOGS]]
 EndUnless


  BRCOMHistStatus nil dlg nil

  BRCOMPopup dlg [0 0] [200 35]

EndScript

;============================================================================
;
;  Called when BRCOMHistStatus, BRCOMHistHist, BRCOMHistUsers has data to
;  display as rows in the dialog application, under the Workbench
;  Application.
;
```

```
;  Inorder to display the data returned as a datalist from the result
;  of the search.
;=========================================================================
DefScript BRCOMHistDisplayRows

  Arguments dlg rows table
  Optionals custom

  Locals errmsg tmplt

   Switch table

     Case :REQUEST

       Unless (Assign tmplt (Send '@wfmgr_create :WFM_Q_TRACKING_TPL) )
         BRComMsg "Can not locate REQUEST template for searching." "BRCOM.VSS" "BRCOMHistDisplayDlg" 2
         Return nil
       EndUnless

       Assign errmsg "No status information available"

     EndCase

     Case :REQHIST

       Unless (Assign tmplt (Send '@wfmgr_create :WFM_Q_TRACKING_HIST_TPL) )
         BRComMsg "Can not locate REQHIST template for searching." "BRCOM.VSS" "BRCOMHistDisplayDlg" 2
         Return nil
       EndUnless

       Assign errmsg "No history information available"

     EndCase

     Otherwise

       Unless (Assign tmplt (Send '@namemgr_create :NAME_GET "TEMPLATE" custom) )
         BRComMsg "Can not locate custom template for searching." "BRCOM.VSS" "BRCOMHistDisplayDlg" 2
```

```
          Return nil
        EndUnless

        ; Ace piece of hacking by RCK
        ; Modify the display method of the date fields so that the time is also displayed

        Assign (Nth 7 (Lookup (Send tmplt :TMPLT_Q_ATTRS) "BROPEN") ) :SYSDSP
        Assign (Nth 8 (Lookup (Send tmplt :TMPLT_Q_ATTRS) "BROPEN") ) '@REQTP_SHOW_DT
        Assign (Nth 7 (Lookup (Send tmplt :TMPLT_Q_ATTRS) "BRCLOSE") ) :SYSDSP
        Assign (Nth 8 (Lookup (Send tmplt :TMPLT_Q_ATTRS) "BRCLOSE") ) '@REQTP_SHOW_DT

        Assign errmsg "No user information available"

      EndCase

   EndSwitch

   Updatedatalist (GetDataList dlg) tmplt dlg t

   SetBuffer dlg rows

   Send (GetDataList dlg) :DL_REFRESH

   If rows Then

     DialogSetItem dlg "msg" ""

   Else

     DialogSetItem dlg "msg" errmsg
     SYS::BEEP

   EndIf


   Return

EndScript
```

```
;=============================================================================
;
; Called from BRIndexSearch - Performs search on indexing database.
;
;
;=============================================================================
DefScript BRComIdxSearch

  Arguments crit

  Locals (count 1) (paramstr "")

  Unless (FBoundP 'BRSQLExecSp)
    LoadScript (VSRootAppend "readdata\\projects\\br\\script\\brsql.vss") :useful
  EndUnless

  DoList (tmp crit)
    If (= count 1) Then
      Assign paramstr (MakeString "@" (First tmp) "=\'" (Second tmp) "\'")
    Else
      Assign paramstr (MakeString paramstr ", @" (First tmp) "=\'" (Second tmp) "\'")
    EndIf
    Assign count (1+ count)
  EndDoList

  Return ( BRSQLExecIdxSchSP  "sp_idxsearch" paramstr)

EndScript

;=============================================================================
;
;  Called from the BRIndexSearch in order to determine the attributes to
;  be used to populate the data panel.
;
;=============================================================================
DefScript BRComIdxAttribs
```

```
Arguments attribs type

Locals  new_attribs base_attribs

  Switch type

    Case 0

    ; CTAX. (Don't want HBIS data)

        Assign new_attribs [ (Lookup attribs "BRCTAXNO" nil :test 'StringEqual) ~
                             ["BRSURNAME" (Lookup attribs "BRCTSURNAME" 2 :test 'StringEqual)   ] ~
                             ["BRFORENAME" (Lookup attribs "BRCTFORENAME" 2 :test 'StringEqual) ] ~
                             ["BRTITLE" (Lookup attribs "BRCTTITLE" 2 :test 'StringEqual)       ] ]
    EndCase

    Case 1

    ; HBIS. (Don't want CTAX data)

      Assign new_attribs [ (Lookup attribs "BRHBISNO" nil :test 'StringEqual) ~
                           (Lookup attribs "BRNINO" nil :test 'StringEqual) ~
                           ["BRSURNAME" (Lookup attribs "BRHBSURNAME" 2 :test 'StringEqual)   ] ~
                           ["BRFORENAME" (Lookup attribs "BRHBFORENAME" 2 :test 'StringEqual) ] ~
                           ["BRTITLE" (Lookup attribs "BRHBTITLE" 2 :test 'StringEqual)       ] ]

    EndCase

    Case 2

    ; BOTH. (Want HBIS and CTAX, use HBIS as basis and then add CTAXNO)

      Assign new_attribs [ (Lookup attribs "BRHBISNO" nil :test 'StringEqual) ~
                           (Lookup attribs "BRNINO" nil :test 'StringEqual)   ~
                           (Lookup attribs "BRCTAXNO" nil :test 'StringEqual) ~
                           ["BRSURNAME" (Lookup attribs "BRHBSURNAME" 2 :test 'StringEqual)   ] ~
                           ["BRFORENAME" (Lookup attribs "BRHBFORENAME" 2 :test 'StringEqual) ] ~
                           ["BRTITLE" (Lookup attribs "BRHBTITLE" 2 :test 'StringEqual)       ] ]
```

```
      EndCase

    EndSwitch

  ; Include standard attribs

  Assign base_attribs ~
  [ (Lookup attribs "BRADDRESS1" nil :test 'StringEqual) ~
    (Lookup attribs "BRADDRESS2" nil :test 'StringEqual) ~
    (Lookup attribs "BRADDRESS3" nil :test 'StringEqual) ~
    (Lookup attribs "BRADDRESS4" nil :test 'StringEqual) ~
    (Lookup attribs "BRPOSTCODE" nil :test 'StringEqual) ~
    (Lookup attribs "BRUPRN"     nil :test 'StringEqual) ~
    (Lookup attribs "BRTEAM"     nil :test 'StringEqual) ~
    (Lookup attribs "BRWKGRP"    nil :test 'StringEqual) ]

  ; combine and return

  Return (Append new_attribs base_attribs)


EndScript

;=============================================================================
;
;  Called from key action bar buttons from within Workbench, Enquiry and
;  Commissioning. Prevents any action from occurring within ViewStar if
;  LGVision has set the disable flag
;
;=============================================================================
DefScript BRCOMButton

  Arguments ctrl abar arg

  Locals ctrlname

  Assign ctrlname (Send ctrl :OBJ_GETPROP '@name)
```

```
Cond
  Case (StringEqual ctrlname "BRWBNEXT")
    BRWorkbenchNext ctrl abar arg
  EndCase

  Case (StringEqual ctrlname "BRWBCLOSE")
    BRFindClose ctrl abar arg
  EndCase

  Case (StringEqual ctrlname "BRWBCOMP")
    BRComplt ctrl abar arg
  EndCase

  Case (StringEqual ctrlname "BRWBFIND")
    BRFind ctrl abar arg
  EndCase

  Case (StringEqual ctrlname "BRWBSEARCH")
    BRIndexSearch ctrl abar arg
  EndCase

  Case (StringEqual ctrlname "BRWBREFER")
    BRReferFromWorkbench ctrl abar arg
  Endcase

  Case (StringEqual ctrlname "BRWBREJECT")
    BRRejectFromWorkbench ctrl abar arg
  EndCase

  Case (StringEqual ctrlname "BRDYGET")
    DYGetDiary ctrl abar arg
  EndCase

  Case (StringEqual ctrlname "BRDYSEND")
    DYSetupDiary ctrl abar arg
  EndCase
```

```
    Case (StringEqual ctrlname "BRMAPTO") ;BW6 todo comment out
      BRLGVMapTo ctrl abar arg ;BW6 todo comment out
    EndCase ;BW6 todo comment out


    Case (StringEqual ctrlname "BRSWAP")
      BRLGVPanelSwitcher
    EndCase

    Case (StringEqual ctrlname "BRCMNEXT")
      BRCMNext
    EndCase

    Case (StringEqual ctrlname "BRWBATTRS")    ;AB1.2 added
      BRWorkBenchUpdate                        ;AB1.2 added
    EndCase                                    ;AB1.2 added
  EndCond
EndScript


;=========================================================================
;
;  Called from Folder Attributes item on the Special menu.
;
;=========================================================================
DefScript BRCOMFolderAttributes

  MoreArguments ignore

  Locals dlg (dlgfile (VSRootAppend "readdata\\projects\\br\\dialogs\\brfldatt.cp")) pan attrs fields

  Unless (And (FBoundP 'CSGetVal) (FBoundP 'CSPutVal))
      LoadScript (VSRootAppend "PROJECTS\\COMMON\\CSDATOOL.FSL") :useful ; BW9 Added
  EndUnless

  Unless (Assign dlg (CSGetVal :BRFLDATTDLG [[:TARGET (GetCurrentEnvir)][:TOPIC :DIALOGS]]))
```

```
    Unless (Assign dlg (OBjLoad dlgfile) )
       BRComMsg "Folder attribute dialog not found" "BRCOM.VSS" "BRCOMFolderAttribs" 2
       Return nil
    EndUnless

    CSPutVal :BRFLDATTDLG dlg [[:TARGET (GetCurrentEnvir)][:TOPIC :DIALOGS]]

  EndUnless

  Unless (GetCurrentAbstract :folder t)
    BRComMsg "No folder currently open" "BRCOM.VSS" "BRCOMFolderAttributes" 2
    Return nil
  EndUnless

  Assign pan (Send dlg :PRLST_Q_PANEL "data")

  Assign attrs (GetDocumentInfo (GetCurrentAbstract :folder t) :attribs)
  Assign fields (Send pan :DPROC_Q_DEF_LIST)

  DoList (fld fields)

     Send pan :DPROC_S_STRING (First fld) (Lookup attrs (First fld) 2 :test 'StringEqual)

  EndDoList

  BRCOMPopup dlg

EndScript


;=========================================================================
;
;  Called from the environement to display the Help Display.
;
;=========================================================================
DefScript BRCOMHelp

MoreArguments ignore
```

```
Locals Path

 Assign Path ["K:\\RR\\VS_BIN\\BRENTHEL"]

 SYS::BEEP ;BW8
 BRCOMmsg "Under construction" "BRCOM.VSS" "BRCOMHelp" ;BW8
;BW8 Send (GetCurrentEnvir) :ENV_HELP "K:\\RR\\VS_BIN\\BRENTHEL.HLP" "Contents"

Return t

EndScript


;=============================================================================
;
;  Called from the environement to position the Dialog Windows in the
;  appropriate position on the Screen.
;
;=============================================================================
DefScript BRCOMPopup

  Arguments dlg
  Optionals (rpos [0 0]) (rsize [100 75])
  Locals (env (GetCurrentEnvir))

  Send env :OBJ_SETPROP :POPUP_REGION_POS rpos
  Send env :OBJ_SETPROP :POPUP_REGION_SIZE rsize

  Return (Popup :alert dlg)

EndScript

;=============================================================================
;
;  Version of ShowMessageBox which calls BRCOMPopup which allows us to
;  control the positioning of message boxes as well as dialog boxes.
;
;=============================================================================
```

```
DefScript BRCOMShowMessageBox

  MoreArguments arglst

  Locals title msglst icon buttons icopict~
         dlg height width buttonstr (xpos 3)(spacer 7) ~
         (minheight 11) (minwidth 40) (xline 10) (yline 4) ~
         (maxline 0)

  Assign title (Or (Second (Member :title arglst)) (Second (Member :TITLE arglst)) "")
  Assign msglst (Or (Second (Member :text arglst)) (Second (Member :TEXT arglst)) nil)
  Assign buttons (Or (Second (Member :buttons arglst)) (Second (Member :BUTTONS arglst)) "")
  Assign icon (Or (Second (Member :icon arglst)) (Second (Member :ICON arglst)) nil)

  Assign buttonstr (MakeString buttons)

  ; Determine height of dialog from min height and no. lines

  Assign height (MAX minheight ( + (Length msglst) 8) )

  ; Determine width of dialog from min width and max message line length

  DoList (msg msglst)
    Assign maxline (MAX maxline (Length msg))
  EndDoList

  Assign width (MAX minwidth (+ maxline 25) )

  ; Create base dialog

  Assign dlg ( NewDialog :title title  :noadd t :h height :w width)


  New3dBox :proc dlg :height 3.15 :width (- width 2)                              ~
           :pos [ ( - height 2.6 ) 2 ]                                            ~
           :name "Bar13d"                                                         ~
           :noadjust t                                                            ~
           :effect :bump                                                          ~
```

```
         :border nil

; determine which buttons should be displayed and in which position

When (StringSearch "YES" buttonstr)

  NewButton :proc dlg :txt "&Yes"                                                ~
            :iconfile ( VSRootAppend "READDATA\\PROJECTS\\BR\\ICONS\\OK.ICO" )   ~
            :pos [ ( - height 2.15 ) xpos ] :name "Yes"                          ~
            :fdata 'BRCOMYes

  Assign xpos (+ xpos spacer)

EndWhen

When (StringSearch "NO" buttonstr)

  NewButton :proc dlg :txt "&No"                                                 ~
            :iconfile ( VSRootAppend "READDATA\\PROJECTS\\BR\\ICONS\\CANCEL.ICO" ) ~
            :pos [ ( - height 2.15 ) xpos ] :name "No"                           ~
            :fdata 'BRCOMNo

  Assign xpos (+ xpos spacer)

EndWhen

When (StringSearch "OK" buttonstr)


  NewButton :proc dlg :txt "&Ok"                                                 ~
            :iconfile ( VSRootAppend "READDATA\\PROJECTS\\BR\\ICONS\\OK.ICO" )   ~
            :pos [ ( - height 2.15 ) xpos ] :name "Ok"                           ~
            :fdata 'BRCOMOK

  Assign xpos (+ xpos spacer)

EndWhen
```

```
When (StringSearch "CAN" buttonstr)


  NewButton :proc dlg :txt "&Cancel"                                              ~
            :iconfile ( VSRootAppend "READDATA\\PROJECTS\\BR\\ICONS\\CANCEL.ICO" )      ~
            :pos [ ( - height 2.15 ) xpos ]  :name "cancel" :cancel t :fdata 'BRCOMCancel

  Assign xpos (+ xpos spacer)

EndWhen


New3dBox :proc dlg :height ( - height 5 ) :width (- width 2)                    ~
         :pos [ 2 2 ]                                                            ~
         :name "DL3d"                                                              ~
         :noadjust t                                                               ~
         :border nil                                                               ~
         :effect :dip

; Display required icon

Cond

  Case (Equal icon :QUEST)
    Assign icopict "help"
  EndCase

  Case (Equal icon :EXCLAIM)
    Assign icopict "singdoc!"
  EndCase

  Case (Equal icon :STOP)
    Assign icopict "trffc14"
  EndCase

  Otherwise
    Assign icopict nil
  EndCase
```

```
  EndCond

  When Icopict

    NewLabel :proc dlg :pos [( - (/ height 2) 1) 3] :picture icopict

  EndWhen

  ; display message text lines

  DoList (msg (Reverse msglst))

    NewField :proc dlg :pos [yline xline] :txt msg :width (+ (Length msg) 5) :noedit t

    Assign yline (+ yline 1)

  EndDoList

  ; popup

  Return (BRCOMPopup dlg)

EndScript


DefScript BRCOMOk
  MoreArguments ignore

  ReturnFromPopup :OK

EndScript


DefScript BRCOMYes
  MoreArguments ignore

  ReturnFromPopup :YES
```

```
EndScript


DefScript BRCOMNo
  MoreArguments ignore

  ReturnFromPopup :NO

EndScript


DefScript BRCOMCancel
  MoreArguments ignore

  ReturnFromPopup :CANCEL

EndScript

;=============================================================================
; Decrypts passwords
;   Argument - encryted password
;   Returns  - decrypted password
;=============================================================================
DefScript BRCOMDecryptPassword

  Arguments passwd

  Locals decpasswd

  Assign decpasswd (@string_decrypt passwd)

  Return decpasswd

EndScript
;=============================================================================
; Encrypts passwords
;   Argument - decrypted password
;   Returns  - encryted password
```

```
;============================================================================
DefScript BRCOMEncryptPassword

  Arguments passwd

  Locals encpasswd

  Assign encpasswd (@string_encrypt passwd)

  Return encpasswd

EndScript
;============================================================================
; Gets user password
;   Argument - user ID
;   Returns  - encryted password
;============================================================================
DefScript BRCOMGetPassword

; Returns encrypted password for the specified user

  Arguments user

  Locals passwd sqlcmd

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

  Assign passwd ( BRSQLExecSP "sp_bruserpassword" [user] )

  Return passwd

EndScript

;============================================================================
;
; Called when the user selects the delete document option from the special
```

```
; menu in Workbench. Only allows deletion of File Notes and Word Docs.
;
; Arguments - Ignored
;
;============================================================================
DefScript BRCOMDelUserDocs
Use "View"                          ;CCS4 Added
Use "Security"
MoreArguments ignore
; Locals (abs (GetCurrentAbstract))  (fvdoc (GetCurrentViewDoc :folder t) ) (absvdoc (GetCurrentViewDoc))
worddocs                       ;CCS4 Commented out
Locals (abs (GetCurrentAbstract))  (fvdoc (GetCurrentViewDoc :folder t) ) (absvdoc (GetCurrentViewDoc)) worddocs
viewpan viewabar cbox curruser ~ ;CCS4 Added
       user owner ret

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

  Unless ( Assign worddocs (CSGetVal :BRWORDDOCLIST [[:TARGET (GetCurrentEnvir)][:TOPIC :CONFIG_VALS]]))

    Unless (Assign worddocs (BRSQLExecSP "SP_BRSelWordDocs"))
      BRComMsg "No Word Documents" "BRCOM.VSS" "BRCOMDelUserDocs" 1
      Return nil
    EndUnless

    CSPutVal :BRWORDDOCLIST worddocs [[:TARGET (GetCurrentEnvir)][:TOPIC :CONFIG_VALS]]

  EndUnless

  Unless (GetCurrentAbstract :folder t)
    BRCOMMsg "There is no folder currently open" "BRCOM.VSS" "BRCOMDelUserDocs" 3
    SYS::BEEP
    return nil
  EndUnless

  When (IsFolder abs)
     BRComMsg "You cannot delete the folder" "BRCOM.VSS" "BRCOMDelUserDocs" 3
```

```
     SYS::BEEP
     Return nil
  EndWhen

  When (= (Length (SelectFromFolder fvdoc :all t)) 1)
     BRComMsg "You cannot delete the only document in the folder" "BRCOM.VSS" "BRCOMDelUserDocs" 3
     SYS::BEEP
     Return nil
  EndWhen

; Unless (Or (StringSearch "File Note" (GetDocumentInfo abs :field "BRDOCTYPE") :test 'StringEqual) ~
;             (Member (GetDocumentInfo abs :field "BRDOCTYPE") worddocs :test 'BRCOMCheckList)         )
;    BRComMsg "You can only delete File Notes or Word Documents from the folder" "BRCOM.VSS" "BRCOMDelUserDocs"
3
;    SYS::BEEP
;    Return nil
; EndUnless

; Unless (Member (GetDocumentInfo abs :field "BRDOCTYPE") worddocs :test 'BRCOMCheckList)
;    BRComMsg "You can only delete Word Documents from the folder" "BRCOM.VSS" "BRCOMDelUserDocs"  3
;    SYS::BEEP
;    Return nil
; EndUnless

  Assign curruser (First(WhoAmI))
  Unless (Or (And (StringSearch "File Note" (GetDocumentInfo abs :field "BRDOCTYPE") :test 'StringEqual) ~
                  (StringEqual curruser "sysadm")) ~
             (Member (GetDocumentInfo abs :field "BRDOCTYPE") worddocs :test 'BRCOMCheckList))
;     BRComMsg "You can only delete Word Documents from the folder.  File notes can only be deleted by System
Administrators." "BRCOM.VSS" "BRCOMDelUserDocs"  3
;     Message ["You can only delete Word Documents from the folder.  File notes can only be deleted by System
Administrators."]
;; ================= Tedy Shalev - Nov 2004 ... ================================
      BRCOMShowMessageBox :title "Delete User Document" ~
                          :text ["Unable to delete this document." ~
                                 "" "If you must be delete this document, please contact the System
Administrator." ""] ~
                             :BUTTONS :OK :ICON :EXCLAIM
```

```
     SYS::BEEP
     Return nil
  EndUnless

;; ============== TEDY - Nov 2004 ... ========================================
  ;; -- Only the original OWNER (who created the Word document) or the SYSADM
  ;;    may delete a Word document that was added to a folder...
  Assign owner (Send (GetCurrentAbstract) :OBJ_GETPROP :OWNER)
  When owner
    ;; -- Old documents (before today) that do not have ownership
    ;;    may be deleted freely by users...
    If (StringEqual owner "SYSADM")
      Then
      Unless (StringEqual owner curruser)
        BRCOMShowMessageBox :title "Delete User Document - by a System Administrator" ~
                            :text ["Unable to delete this document." ~
                                   "" (MakeString "Only the System Administrator may delete this document!") ""]
~
                            :BUTTONS :OK :ICON :STOP
        SYS::BEEP
        Return
      EndUnless
    Else
      Unless (Or (StringEqual owner curruser) (StringEqual "SYSADM" curruser))
        BRCOMShowMessageBox :title "Delete User Document" ~
                            :text ["Unable to delete this document." ~
                                   "" (MakeString "Only " (QueryUser owner) " or the System Administrator may
delete this document!") ""] ~
                            :BUTTONS :OK :ICON :STOP
        SYS::BEEP
        Return
      EndUnless
    EndIf
  EndWhen
;; ================== ... Tedy Shalev - Nov 2004 ================================

  When (Assign absvdoc (First (SelectFromFolder fvdoc :itemnums [(QueryCurrentDocument fvdoc)])) )
;; ============== TEDY - Nov 2004 ... ========================================
```

```
    ;; -- Commented out - unfriendly message box, replaced with neater one :-)
    ;Unless (Equal (ShowMessageBox :title "Delete Document" :text ["Do you really wish to delete the current
document ?"] :buttons :YESNO :icon :QUEST) :YES)
    ;  Return
    ;EndUnless

    Assign ret (BRCOMShowMessageBox :title "Delete User Document" ~
                         :text ["" ~
                                "Do you really wish to delete the current document ?" ""] ~
                         :BUTTONS :YESNO :ICON :QUESTION)
    Unless (Equal ret :YES)
      Return
    EndUnless
;; ================= ... Tedy Shalev - Nov 2004 ================================

    DeleteFromFolder fvdoc absvdoc

;   ====================================================================
;   Temporary fix to allow the use of delete page and delete document   ;CCS4 Added
;   functions on 32 bit clients.                                        ;CCS4 Added
;   UpdateDisplay has been replaced with a call to refresh the View     ;CCS4 Added
;   action panel TOCBOX. There is still a call outstanding with VS to   ;CCS4 Added
;   get UpdateDisplay to work on 32 bit clients.                        ;CCS4 Added
;   ====================================================================
;   Now working in ViewStar 5.00 - Tedy - 28-Oct-1999 !
;   ====================================================================
    UpdateDisplay                                                       ;CCS4 Commented out

;   Assign viewpan   ( GetPanel "VIEW" )                                ;CCS4 Added
;   Assign viewabar  ( GetActionbar "VIEW" )                            ;CCS4 Added
;   Assign cbox (Send viewabar :DIALOG_Q_CTRL "BRWBTOCBOX")             ;CCS4 Added
;   TocSelect cbox viewabar nil                                         ;CCS4 Added

  EndWhen

EndScript
;=============================================================================
;
```

```
; Custom Test function for Member function
;
;    Arguments arg1 - string
;              arg2 - list with a single item
;
;=============================================================================
DefScript BRCOMCheckList

  Arguments arg1 arg2

  If ( Equal arg1 ( First arg2 ) ) Then
    Return t
  Else
    Return nil
  EndIf

EndScript
;************************
; AB 1.3 Added below code
;************************
;=============================================================================
;
; Calculates Target Dates given the Start date and the interval
;
;    Arguments from_date - date in VS internal Format or as string in "dd/MM/yy" format
;              interval  - no of working days to add to the start date
;
;=============================================================================
DefScript BRCOMTargetDate

  Arguments from_date interval

  Locals  to_date from_day remainder quotient tot_days ~
          (buffer [ [4 ["Tue" "Wed" "Thu" "Fri"] ] ~
                    [3 ["Wed" "Thu" "Fri"] ]       ~
                    [2 ["Thu" "Fri"] ]             ~
                    [1 ["Fri"] ]                   ~
                  ] )
```

```
; Convert the from_date to VS Internal Format if necessary

  If ( Type from_date '@cons ) Then
    Assign from_date (BrComInternalDate from_date)   ;---BW1 Added---
  Else
       Assign from_date (BrComInternalDate from_date)  ; ---BW1 Added---
  EndIf

; Check date and interval are valid before continuing

  Unless (And from_date interval)
    Return nil
  EndUnless

; To calculate the target date we need to know
; i)  the number of whole weeks contained in the interval, do this by dividing the interval by 5 (the number of
days in a working week)
; ii) the number of additional days , i.e a fractional part of a working week expressed in days. Calculated by
finding the remainder of the interval divided by 5
; e.g an interval of 23, contains 4 whole working weeks and an additional 3 days

; This is the number of additional days
  Assign remainder (MOD interval 5)

; This is the number of whole weeks contained in the interval
  Assign quotient (/ (- interval remainder) 5)

; Depending on the day of the week that the from_date falls and the number of additional days,
; the additional days may span over a weekend

; Get the day of the week for from_date
  Assign from_day (FormatDateTime from_date "ddd" nil)

; AB1.5 - Added Code

; If the from_date is a Saturday we need to start our calculation from the Monday that follows
  When (StringEqual from_day  "Sat")
```

```
    Assign from_date (DateTime+ from_date 2 :day)
    Assign from_day  (FormatDateTime from_date "ddd" nil)
  EndWhen

; If the from_date is a Suny we need to start our calculation from the Monday that follows
  When (StringEqual from_day  "Sun")
    Assign from_date (DateTime+ from_date 1 :day)
    Assign from_day  (FormatDateTime from_date "ddd" nil)
  EndWhen

; AB1.5 - End of Added Code

; Then given the remaining number of days, and the day of the week on which the from
; date fell we can work out if any of the remaining days are part of a weekend.

  If (Member from_day (Lookup buffer remainder 2) :test 'StringEqual)  Then
;   additional days span over a weekend
    Assign tot_days (+ (* quotient 7) (+ remainder 2) )
  Else
;   additional days DON'T span over a weekend
    Assign tot_days (+ (* quotient 7) remainder)
  EndIf

  Assign to_date (DateTime+ from_date tot_days :day)

  Return to_date

EndScript
;************************
; AB 1.3 Added above code
;************************

;************************
; AB 1.8 Added below code
;************************
;=========================================================================
;
; Called when doctypes required to populate Work Description Dialog
```

```
;
;    Arguments - wktype
;
;=========================================================================
DefScript BRComGetDocTypes

  Arguments wktype

  Locals doctypes

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

  Unless (Assign doctypes (CSGetVal :BRDOCTYPES [[:TARGET (GetCurrentEnvir)][:TOPIC :CONFIG_VALS]]))

    ; Get List Of Valid Worktypes
    Unless (FBoundP 'BRSQLExecSp)
      LoadScript (VSRootAppend "readdata\\projects\\br\\script\\brsql.vss") :mandatory
    EndUnless

    Unless (Assign doctypes (BRSQLExecSP  "sp_brwktypedoctypes" [wktype] ))
      BRComMsg "Worktypes not retrieved." "BRCOM.VSS" "BRComGetDocTypes" 1
      Return nil
    EndUnless

    CSPutVal :BRDOCTYPES doctypes [[:TARGET (GetCurrentEnvir)][:TOPIC :CONFIG_VALS]]
  EndUnless

  Return doctypes

EndScript

;=========================================================================
;
; Called when doctypes required to populate Work Description Dialog
;
;    Arguments - None
```

```
;
;=========================================================================
DefScript BRCOMSelectWkDescOK

  Arguments ignore dlg ignore

  Locals wkdescdl

  Assign wkdescdl (GetDataList dlg "wkdescdl")

  Unless (Send wkdescdl :DL_Q_ROW)
    DialogSetItem dlg  "Msg" "You must select an item" nil
    Return nil
  EndUnless


  ReturnFromPopup (First (First (Send wkdescdl :DL_Q_ROW)))

EndScript

;************************
; AB 1.8 Added above code
;************************




;=========================================================================
;
;    Author   Mayank Ladd
;    Function BRComMirTrkDoc
;    Version  BW12
;    Arguments - fname vdoc
;    Optionals - fabs queuein brastart
;
;=========================================================================

DefScript BRComMirTrkDoc
```

```
Arguments fname vdoc
Optionals fabs queuein brastart
Locals startdate heldnodays brladate vdoclist

   ; Load neccessary script files
   Unless ( fboundp 'BRComMsg )
     LoadScript ( VSRootAppend "READDATA\\PROJECTS\\BR\\SCRIPT\\BRCOM.VSS" ) :useful
   EndUnless

   Cond


;=============================================================================================================
====================================
      ; CASE 1. Set the assessment start date to batch date for all new documents scanned or imported into the
system.
      ;   NOTE. All new folders and Documents pass through the 'BRImportSetCustomProps' function in BRIMPORT.VSS

;=============================================================================================================
====================================
      Case (StringEqual fname "BRImportSetCustomProps")

            Assign brastart (GetDocumentInfo vdoc :field "BRBATDATE")

        EndCase


;=============================================================================================================
====================================
      ; CASE 2. Add period held by Co to document assessment date if folder is reject or referred from the Client
Officer application.
      ;
      ;   NOTE. This function is only called when a folder is referred or reject from the 'Client Officer'
application back to the 'WorkBench' application.

;=============================================================================================================
====================================
      Case (OR (StringEqual fname "BRRejCom") (StringEqual fname "BRRefer"))
```

```
          ;; Need to get open and close dates from the folder
         Unless fabs
            BRComMsg "Folder Level Information Required." "BRCOM.VSS" "BRComMirTrkDoc" 1
            Return nil
         EndUnless

         ; We use the last action date date because this is the date the folder was forwarded from Workbench to
         ; the client officer application.

         Unless (Assign brladate (GetDocumentInfo fabs :Field "BRLADATE"))
            BRComMsg "Folder Level Information Required BRLADATE." "BRCOM.VSS" "BRComMirTrkDoc" 1
            Return nil
         EndUnless

         If (Type brladate '@String) Then
            Assign brladate (BrComInternalDate brladate)
         EndIf

         ; Add days held to start date of document

         Assign startdate (BrComInternalDate (GetDocumentInfo vdoc :field "BRASTART"))
         Assign heldnodays (DateTime- (GetCurrentDateTime) brladate :day)
         Assign brastart (datetime+ startdate heldnodays :day)
         Assign brastart (BrComDateString brastart)

       EndCase


;=============================================================================================================
===================================
     ; CASE 3. Folder expires from diary and has been in the diary
     ;         for greater than 24 hours

;=============================================================================================================
===================================
       Case (StringEqual fname "DYDiaryResetDocAssStartDate")
```

```
          Assign brastart (GetCurrentDateTime)
          Assign brastart (BrComDateString brastart)

        EndCase


;=============================================================================================================
====================================
      ; Otherwise

;=============================================================================================================
====================================
        Otherwise

           Return nil

        EndCase

   EndCond

   Return brastart

EndScript


;=========================================================================
;Constuction.
;=========================================================================

DefScript BRComUnderCon

   MoreArguments ignore

   BRCOMShowmessagebox :title "System Warning Message..." :text ["Not available. Under Construction"] :BUTTONS
:OK :ICON :EXCLAIM

EndScript
```

```
;; -------------------------------------------------------------------------
;; 10-March-1999
;; Tedy
;; Added on following script...
;-------------------------------------------------------------------------
;  BRSetUserSearchCriteria
;
;  DESCRIPTION:
;    Get UPRN range as assigned to a given user (by user ID)
;
;  ARGUMENTS:
;    none
;
;  RETURNS:
;    List containing first UPRN and last UPRN assigned to this user.
;-------------------------------------------------------------------------
DefScript BRSetUserSearchCriteria
Arguments userID
Optionals phase
Locals    (env (GetCurrentEnvir)) uprnList fromUPRN toUPRN subgroup ~
          grp cbUser cbTeam mainBar (HBTeams ["White" "Green" "Red" "Blue"])
Use "Security"

   ;; Unable to carry on if userID is not passed in here...
   Unless (Or userID (IsUserInGroup userID "SYSADM") (IsUserInGroup userID "CS"))
     Return
   EndUnless

   ;; Cannot carry-on unless a user and a team are selected in
   ;; the combo-boxes on the main action bar...
   Assign mainBar (GetActionBar "Main")
   Assign cbUser  (Send mainBar :DIALOG_Q_CTRL "BRUSER")
   Assign cbTeam  (Send mainBar :DIALOG_Q_CTRL "BRINBOX")
   Assign grp (Second (Send cbTeam :CBOX_Q_ROW (Send cbTeam :CBOX_Q_PICK)))

   ;; Get the UPRN range for this user id...
```

```
Unless (Fboundp 'BRGetUPRN_UserAssignment)
  LoadScript (VSRootAppend "readdata\\projects\\br\\script\\brassign")
EndUnless

Assign uprnList (BRGetUPRN_UserAssignment userID)
;; Ensure that this user does have an assignment, else leave it opened to ALL UPRNs...
Assign fromUPRN (First  uprnList)
Assign toUPRN   (Second uprnList)
;; Ensure that fields that are read as "NIL" from the database tables
;; are interpreted correctly, ie. as nil value!
When (StringEqual fromUPRN "NIL")
  Assign fromUPRN nil
EndWhen
When (StringEqual toUPRN "NIL")
  Assign toUPRN nil
EndWhen

Assign subgroup (Third  uprnList)
If (And fromUPRN toUPRN)
   Then
     ;;   :BRPROCESSMODE     - process mode, set up in custom data.
     ;Unless (CsGetVal :BRPROCESSMODE [[:TARGET env] [:TOPIC :CONFIG_VALS]])
        CSPutVal :BRPROCESSMODE [                              ~
                                   [:PROCMODE :ASSIGNED]     ~
                                   [:WORKTYPE nil]           ~
                                   [:WFSTATUS "Active"]      ~
                                   [:UPRN fromUPRN toUPRN]]  ~
                                 [                           ~
                                   [:TARGET env]             ~
                                   [:TOPIC :CONFIG_VALS]     ~
                                 ]
     ;EndUnless
Else
        CSPutVal :BRPROCESSMODE [                              ~
                                   [:PROCMODE :ASSIGNED]     ~
                                   [:WORKTYPE nil]           ~
                                   [:WFSTATUS "Active"]]     ~
                                 [                           ~
```

```
                                    [:TARGET env]             ~
                                    [:TOPIC :CONFIG_VALS]     ~
                                  ]

        ;; If current user is in one of the Housing Benefits (colour based team)
        ;; issue a warning.    Unless the user belongs to subgroup 1 or 2, being
        ;; the Team Leader or the Deputy Team Leader...
        When (Member grp HBTeams)
          ; Load neccessary script files...
          Unless (fboundp 'BRCOMShowmessagebox)
            LoadScript (VSRootAppend "READDATA\\PROJECTS\\BR\\SCRIPT\\BRCOM")
          EndUnless
          BRCOMShowMessageBox :title "User Work Assignment Error" ~
                              :text [(MakeString "User " userID " was not assigned an UPRN range!") ~
                                     "" "Please contact your Team Leader."  ""] ~
                              :BUTTONS :OK :ICON :STOP
        EndWhen
    EndIf
    Return uprnList

EndScript ;BRSetUserSearchCriteria
```