
The Utilities Library

1 AgentLogEntry [Procedure]

Version: 2.x to 4.7
Availability: Word, Excel, Text, Export, Import

AgentLogEntry (EntryType As Integer, EntryText As String)

This procedure creates an entry in the Integra agent log database. The procedure can be used for debugging purposes and is particularly useful for scheduled or web-browser driven profiles.

Parameters:

EntryType (Integer)

0 (zero) indicates a simple log message.

1 indicates a runtime error.

EntryText (String)

Message to be shown in the log.

Example:

```
Dim ws As NotesUIWorkspace
```

```
If ws Is Nothing Then
```

```
Call AgentLogEntry(1,"The Notes Workspace object cannot be set")
```

```
End If
```

2 ArrayFormat [Function]

Version: 2.x to 4.7
Availability:

ArrayFormat (Item As Variant) As Variant

This function is calling the FormatSingleValue() function. However, ArrayFormat will accept single values as well as arrays (even of mixed data types). In case an array is passed on, it will call the FormatSingleValue() function for each array element.

Parameters:

Item (Variant)

Single value or array of the following data types:

Return Value:

Variant array of strings.

Example:

[Internal use for Integra's collection sorting. No example available.](#)

3 ArrayToString [Function]

Version: 2.x to 4.7
Availability:

ArrayToString (ItemArr As Variant, Separator As String) As String

This function converts an array of items into a text string, each array element separated by the separator string.

Parameters:

ItemArr (Variant)

Array to be converted into a string. The array's elements can consist of the following data types: text, date time, integer, long, single or double. Arrays of mixed data types are allowed.

Separator (String)

Single character or a text string used as separator.

Return Value:

String.

Example:

```
Dim V_array As Variant
Dim S_Array(2) As String
Dim S_Text As String

S_Array(0) = "Car"
S_Array(1) = "Airplane"
S_Array(2) = "Train"

V_Array = S_Array

S_Text = ArrayToString(V_Array, ";")

Call MsgBox(S_Text, 0, "ArrayToString Result")
```

4 AttachFile [Function]

Version:

Availability:

AttachFile (Note As NotesDocument, Filename As Variant, SaveWithName As Variant, RItem As NotesRichTextItem, ReAttach As Integer, SaveDoc As Integer, KillFile As Integer, ShowPBar As Integer) As Integer

This function is being replaced by the function EmbedFile() and is kept for compatibility reasons. In matter of fact it is calling EmbedFile() with an additional fixed parameter. Please refer to the function EmbedFile().

5 CheckRights [Function]

Version:

Availability:

CheckRights (Role As String) As Integer

This function checks whether a particular user role has been assigned to the current user's ACL entry of the current database.

Parameters:

Role (String)

The role to be checked.

Return Value:

Integer. True or False.

Example:

```
If Not CheckRights("[Administrator]") then
    Call MsgBox("You do not have sufficient execution rights.", 0, "Warning!")

Exit Sub
End If
```

6 ClearIntegraTempDir [Procedure]

Version:
Availability:

Sub ClearIntegraTempDir

Parameters:

Return Value:

Example:

```
' Callback Script Example
Const CB_INITIALISE = 3
Const CB_BFORREADNOTES = 6
Const CB_BFORWRITECOM = 1
Const CB_AFTERWRITECOM = 9
Const CB_BFORCOMACTION = 5
Const CB_BFORNOTESACTION = 2
Const CB_TERMINATE = 4

select case stats.cbstatus
case CB_INITIALISE
case CB_BFORREADNOTES
case CB_BFORWRITECOM
case CB_AFTERWRITECOM
case CB_BFORCOMACTION
case CB_BFORNOTESACTION
case CB_TERMINATE
end select
```

7 CreateMemo [Function]

Version:
Availability:

CreateMemo (SendTo() As String, CopyTo() As String, Bcc() As String, Subject As String, BodyHdr As String, Filename As Variant, ConfirmDlg As Integer, AppendRtitem As NotesRichTextItem, KillFile As Integer, ShowPBar As Integer, Note As NotesDocument) As Integer

This function is used by Integra's runtime libraries to create a new memo document for profiles utilising the Send To option. The function actually receives the blank memo document and just sets all necessary fields on the document and attaches given filenames. Optional it can present a mail-send dialog requesting the user to enter the SendTo, CopyTo, BCC, Subject and Body heading information.

More useful is probably the function SendMail(), which is in matter of fact calling CreateMemo(), but also sends out the mail if CreateMemo() could create a valid (send able) memo document.

Parameters:

All parameters are explained in the function SendMail(). CreateMemo() has only one additional parameter:

Note (NotesDocument)

Blank memo document. This document can be created in either the user's mail database, in the current database (Integra) or any other database. The document is only a temporarily created mail container and is not saved to file.

Return Value:

Integer. True or False.

Example:

See SendMail()

8 CreateSortingString [Procedure]

Version:
Availability:

Sub CreateSortingString

Parameters:

Return Value:

Example:

```
' Callback Script Example
Const CB_INITIALISE = 3
Const CB_BFORREADNOTES = 6
Const CB_BFORWRITECOM = 1
Const CB_AFTERWRITECOM = 9
Const CB_BFORCOMACTION = 5
Const CB_BFORNOTESACTION = 2
Const CB_TERMINATE = 4

select case stats.cbstatus
case CB_INITIALISE
case CB_BFORREADNOTES
case CB_BFORWRITECOM
case CB_AFTERWRITECOM
case CB_BFORCOMACTION
case CB_BFORNOTESACTION
case CB_TERMINATE
end select
```

9 DetachFile [Function]

Version:
Availability:

DetachFile (Note As NotesDocument, Path As String, FileName As Variant, SaveWithName As Variant, All As Integer, ShowPBar As Integer) As Integer

This function will detach one or more files from the passed on Notes document. Both, the Notes back-end document as well as the rich text item has to be declared and set before being passed on to this function.

Parameters:

Note (NotesDocument)

The Notes back-end document the files are attached to.

Path (Variant)

Path in which the to be detached files are created (saved) in.

Filename (Variant)

Filename is a return parameter returning the list of filenames that are detached by the process. The parameter passed on as Filename must be declared as of type Variant().

SaveWithName (Variant)

The name or names with which the file(s) are stored in Path. This parameter can be left blank (empty string) in which case the attachments are saved to the file-system with the same filename as they have been attached with, otherwise it has to correspond with the Filename parameter: if Filename is of type String, this parameter has to be of type String; if the Filename parameter is of type String() then this parameter has to be of type String() and for each element in FileName() one corresponding element in SaveWithName() has to exist.

All (Integer)

True or False. If True, files successfully attached or embedded, will be deleted from the file system.

ShowPBar (Integer)

True or False. If True, this function will display a progress bar.

Return Value:

Integer. True or False.

10 EmbedFile [Function]

Version:

Availability:

EmbedFile (Action As String, Note As NotesDocument, Filename As Variant, SaveWithName As Variant, Ritem As NotesRichTextItem, ReAttach As Integer, SaveDoc As Integer, KillFile As Integer, ShowPBar As Integer) As Integer

This function will embed or attach one or more files to a Notes document. Both, the Notes back-end document as well as the rich text item has to be created previously and passed on to this function.

Parameters:

Action (String)

Either "Attach" or "Embed". When using the "Embed" option this function will create an embedded object within the passed on rich text item. Only one file can be embedded. The "Attach" option allows creating one or more attachments.

Note (NotesDocument)

The Notes back-end document the files are to be attached or embedded to.

Filename (Variant)

Names of the file or files to be attached or embedded. The filename(s) have to include the path of the file. The parameter can either be of type String containing a single filename or of type String() or Variant(), containing an array of filenames.

SaveWithName (Variant)

The name or names with which the file(s) are stored in the Notes document (object name). This parameter can be left blank (empty string) in which case the or embedded object or attachments are stored with the physical filename, otherwise it has to correspond with the Filename parameter: if Filename is of type String, this parameter has to be of type String; if the Filename parameter is of type String() then this parameter has to be of type String() and for each element in FileName() one corresponding element in SaveWithName() has to exist.

Ritem (NotesRichTextItem)

The NotesRichTextItem object in which the file(s) are to be attached or embedded to.

ReAttach (Integer)

True or False. If True, existing attachments with the same filename or same object name (SaveWithName parameter) will be deleted prior to attaching or embedding the new file(s).

SaveDoc (Integer)

True or False. If True the Notes document will be saved after attaching or embedding the file(s).

KillFile (Integer)

True or False. If True, files successfully attached or embedded, will be deleted from the file system.

ShowPBar (Integer)

True or False. If True, this function will display a progress bar.

Return Value:

Integer. True or False.

Example:

```
Dim doc As NotesDocument
Dim ritem As NotesRichTextItem
Dim fnam(0) As Variant
Dim savfnam(0) As Variant
```

```
Set doc = stats.savdoc
Set ritem = doc.GetFirstItem("Attachments")

If ritem Is Nothing Then
    Set ritem = doc.CreateRichTextItem("Attachments")
End If

fnam(0) = "c:\MyDocument.doc"
savfnam(0) = ""

If EmbedFile("Attach",doc,fnam,savfnam,ritem,False,True,False,False) Then
    Call MessageDialog("File Attached/Embedded Successfully",0,"Embed File")
Else
    Call MessageDialog("An error occurred while attaching/embedding the" & _
        "file",0,"Embed File")
End If
```

11 EvaluateFormula [Function]

Version:

Availability:

Function EvaluateFormula (formula As String,doc As NotesDocument) As Variant

Parameters:

Return Value:

Example:

```
' Callback Script Example
Const CB_INITIALISE = 3
Const CB_BFORREADNOTES = 6
Const CB_BFORWRITECOM = 1
Const CB_AFTERWRITECOM = 9
Const CB_BFORCOMACTION = 5
Const CB_BFORNOTESACTION = 2
Const CB_TERMINATE = 4

select case stats.cbstatus
case CB_INITIALISE
case CB_BFORREADNOTES
case CB_BFORWRITECOM
case CB_AFTERWRITECOM
case CB_BFORCOMACTION
case CB_BFORNOTESACTION
case CB_TERMINATE
end select
```

12 ExecuteTextErrorLine [Function]

Version:

Availability:

ExecuteTextErrorLine (ErrorText As String, ProcStr As String) As String

When Integra is executing Lotus Script at runtime that has been stored in the Callback Script field or the Error Handler field (Integra profile), any error event caused by this script will not allow to retrieve the line number in which the code has failed using the Lotus Script function Erl().

The line number cannot be retrieved with the Lotus Script Erl(), as the code is compiled and executed at runtime. In such cases the Erl() function returns 0 (zero) and the Error() function returns a string containing a text sequence \$EXECUTE\$xxx, whereas xxx represents the line number.

The ExecuteTextErrorLine function analysis the error text returned by the function Error() to retrieve the line number and to return the substring of the code line that has failed.

This function could eventually be used if you try to implement your own error handling.

Parameters:

ErrorText (String)

Error text that has been either retrieved using the Error() function or the Integra runtime variable Export.ErrorText or Import.ErrorText (depending on the type of profile executed).

ProcStr (String)

String representing the Lotus Script code that has been executed and caused the error event.

Return Value:

String.

13 ExtractDisk [Function]

Version:

Availability:

ExtractDisk (Filename As String) As String

This function extracts the disk-name component from a given path-file-name.

Parameters:

Filename (Type)

Name of the file including its path.

Return Value:

String.

Example:

```
Dim diskname As String
```

```
filename = ExtractDisk("c:\MyDocuments\Integra.doc")
```

```
Call MessageDialog(diskname,0,"Disk Name")
```

14 ExtractFileName [Function]

Version:

Availability:

ExtractFileName (Filename As String) As String

This function extracts the file-name component (with extension) from a given path-file-name.

Parameters:

Filename (String)

Name of the file including its path.

Return Value:

String.

Example:

```
Dim filename As String
```

```
filename = ExtractFileName("c:\MyDocuments\Integra.doc")
```

```
Call MessageDialog(filename,0,"File Name")
```

15 ExtractPath [Function]

Version:
Availability:

ExtractPath (Filename As String) As String

This function extracts the path-name component from a given path-file-name.

Parameters:

Filename (String)

Name of the file including its path.

Return Value:

String.

Example:

```
Dim pathname As String
```

```
pathname = ExtractPath("c:\MyDocuments\Integra.doc")
```

```
Call MessageDialog(pathname,0,"Path Name")
```

16 FileInUse [Function]

Version:
Availability:

FileInUse (Filename As String) As Integer

This function checks if a particular file located in the file system is in use. It first calls the function `IsMsOfficeRunning(Filename)` which in turn makes some API calls looking for any MS Word or Excel task that may have the specified file in use (see `IsMsOfficeRunning`). If the file is in use by MS Office, the `FileInUse` function returns `True` at this point. Otherwise it will perform a low-level exclusive file access. If successful the file is not in use by any application and `FileInUse` returns `False`, otherwise `True`.

There are situation where MS Office temporarily closes a file that it has in use and therefore a pure low-level file access check might cause unpredictable results. That is why this function checks first for existing MS Office tasks and communicates with the task whether it has a particular file in use or not. This effect is to keep in mind when using this function on other filetypes used by different applications than MS Office, as in such case only the low-level file access will determine the `FileInUse`'s result.

Parameters:

Filename (String)

Name of the file including its path.

Return Value:

Integer. True or False.

Example:

```
Dim fnam As String
```

```
fnam = "c:\MyDocument.doc"
```

```
If FileInUse(fnam) Then
```

```
    Call MessageDialog("The file " & fnam & " is in use",0,"File In Use")
```

```
End If
```

17 FindCharPos [Function]

Version:
Availability:

Function `FindCharPos` (src As String, Byval char As String) As Integer

Parameters:

Return Value:

Example:

```
' Callback Script Example
Const CB_INITIALISE = 3
Const CB_BFORREADNOTES = 6
Const CB_BFORWRITECOM = 1
Const CB_AFTERWRITECOM = 9
Const CB_BFORCOMACTION = 5
Const CB_BFORNOTESACTION = 2
Const CB_TERMINATE = 4

select case stats.cbstatus
case CB_INITIALISE
case CB_BFORREADNOTES
case CB_BFORWRITECOM
case CB_AFTERWRITECOM
case CB_BFORCOMACTION
case CB_BFORNOTESACTION
case CB_TERMINATE
end select
```

18 FormatSingleValue [Function]

Version:
Availability:

FormatSingleValue (AnyValue As Variant) As String

This function is using the Lotus Script format() function with pre-defined format pictures for the different data types:

INTEGER	: "0000000000000000"
LONG	: "00"
SINGLE	: "0000000000000000.0000"
DOUBLE	: "0000000000000000000000000000000000.00000000"
DATE	: "yyymmdd hhnnss"
BOOLEAN	: "0" or "1"
STRING	: "..."

If AnyValue is of type BOOLEAN, the return value is "0" for False and "1" for True.

This function is mainly used for Integra's internal Notes document collection sorting algorithm.

Parameters:

AnyValue (Variant)

The value to be formatted to a string value. Any of the above data types is allowed as single value. Arrays or lists are not allowed.

Return Value:

String.

19 GetIntegraDataDir [Function]

Version:
Availability:

GetIntegraDataDir () As String

This function returns the path to Integra's temporary file directory. If the directory doesn't exist it will create the subdirectory Integra4Notes.Temp in the Notes data directory. In older versions of Integra for Notes this function was called GetNotesDataDir(). For compatibility reasons the GetNotesDataDir() is still available, but should not be used anymore.

Return Value:

String.

Example:

```
Dim integradir As String
Integradir = GetIntegraDataDir()
Call MessageDialog(integradir,0,"Integra Data Directory")
```

20 GetRandomFileName [Function]

Version:**Availability:**

GetRandomFileName (FilePath As String, FilePreFix As String, FileExtention As String, forceRND As Integer) As String

This function will return a random and unique path-file-name.

Parameters:

FilePath (String)

Default path (directory) in which the file will be created.

FilePreFix (String)

Defines a fixed prefix for the random filename.

FileExtention (String)

Defines the extention of the file that will be created.

forceRND (Integer)

True or False. If True, the function will always add a random number to the given FilePreFix, otherwise it will first check if the given parameters result in a valid and unique path-file-name before adding a random number.

Return Value:

String.

Example:

```
Dim fnam As String
fnam = GetRandomFileName("c:\", "Integra", ".txt", False)
Call MessageDialog(fnam, 0, "Random File Name")
```

21 GetSystemMessage [Function]

Version:**Availability:**

Function GetSystemMessage (Byval no As String, icon As Long, title As String) As String

Parameters:**Return Value:****Example:**

```
' Callback Script Example
Const CB_INITIALISE = 3
Const CB_BFORREADNOTES = 6
Const CB_BFORWRITECOM = 1
Const CB_AFTERWRITECOM = 9
Const CB_BFORCOMACTION = 5
Const CB_BFORNOTESACTION = 2
Const CB_TERMINATE = 4
```

```
select case stats.cbstatus
case CB_INITIALISE
case CB_BFORREADNOTES
case CB_BFORWRITECOM
case CB_AFTERWRITECOM
case CB_BFORCOMACTION
case CB_BFORNOTESACTION
case CB_TERMINATE
end select
```

22 GetUserProfile [Function]

Version:
Availability:

GetUserProfile () As Notesdocument

This function returns an object handle (NotesDocument) to the user's personal profile created in the Integra database. The very first time a user is running an Integra import or export a personal user profile is created in the Integra database to store temporary runtime information.

The user profile can be utilised to temporarily store variables between the different event calls of the Callback Script. Refer also to SetUserProfileVar() and GetUserProfileVar()

Return Value:

NotesDocument object handle.

Example:

```
Dim prof As NotesDocument
Set prof = GetUserProfile()

If prof Is Nothing Then
    Call MessageDialog("An error has occurred while getting the user profile", 0, "Get User Profile")
End If
```

23 GetUserProfileVar [Function]

Version:
Availability:

GetUserProfileVar (VarName As String, Default As Variant) As Variant

This function returns the value (previously stored) in a field VarName in the user's personal profile.

The following data types are allowed: INTEGER, LONG, SINGLE, DOUBLE, DATETIME, BOOLEAN, STRING.

Tip: If you need to temporarily store a reference to a Notes document you can do this by saving the document's UNID and retrieve back the handle using the NotesDatabase.GetDocumentByUNID() method.

Parameters:

VarName (String)
Name of the variable (field) which contents should be returned.

Default (Variant)
If no value has previously stored, return this default value.

Return Value:

Variant.

Example:

```
Call SetUserProfileVar("ClientName","Mr John Smith")

Dim client As String

Client = GetUserProfileVar("ClientName", "")

Call MessageDialog(client,0,"Client Name")
```

24 GetWebActivityDoc [Function]

Version:
Availability:

Function GetWebActivityDoc(unid As String) As NotesDocument

Parameters:

Return Value:

Example:

```
' Callback Script Example
Const CB_INITIALISE = 3
Const CB_BFORREADNOTES = 6
Const CB_BFORWRITECOM = 1
Const CB_AFTERWRITECOM = 9
Const CB_BFORCOMACTION = 5
Const CB_BFORNOTESACTION = 2
Const CB_TERMINATE = 4

select case stats.cbstatus
case CB_INITIALISE
case CB_BFORREADNOTES
case CB_BFORWRITECOM
case CB_AFTERWRITECOM
case CB_BFORCOMACTION
case CB_BFORNOTESACTION
case CB_TERMINATE
end select
```

25 IsDateTimeValue [Function]

Version:
Availability:

Function IsDateTimeValue (datetime As Variant) As Integer

Parameters:

Return Value:

Example:

```
' Callback Script Example
Const CB_INITIALISE = 3
Const CB_BFORREADNOTES = 6
Const CB_BFORWRITECOM = 1
Const CB_AFTERWRITECOM = 9
Const CB_BFORCOMACTION = 5
Const CB_BFORNOTESACTION = 2
Const CB_TERMINATE = 4

select case stats.cbstatus
case CB_INITIALISE
case CB_BFORREADNOTES
case CB_BFORWRITECOM
case CB_AFTERWRITECOM
case CB_BFORCOMACTION
case CB_BFORNOTESACTION
case CB_TERMINATE
end select
```

26 IsUserProfileLocked [Function]

Version:
Availability:

Function IsUserProfileLocked () As Integer

Parameters:

Return Value:

Example:

```
' Callback Script Example
Const CB_INITIALISE = 3
Const CB_BFORREADNOTES = 6
Const CB_BFORWRITECOM = 1
Const CB_AFTERWRITECOM = 9
Const CB_BFORCOMACTION = 5
Const CB_BFORNOTESACTION = 2
Const CB_TERMINATE = 4

select case stats.cbstatus
case CB_INITIALISE
case CB_BFORREADNOTES
case CB_BFORWRITECOM
case CB_AFTERWRITECOM
case CB_BFORCOMACTION
case CB_BFORNOTESACTION
case CB_TERMINATE
end select
```

27 LockUserProfile [Procedure]

Version:
Availability:

Sub LockUserProfile(cLock As String)

Parameters:

Return Value:

Example:

```
' Callback Script Example
Const CB_INITIALISE = 3
Const CB_BFORREADNOTES = 6
Const CB_BFORWRITECOM = 1
Const CB_AFTERWRITECOM = 9
Const CB_BFORCOMACTION = 5
Const CB_BFORNOTESACTION = 2
Const CB_TERMINATE = 4

select case stats.cbstatus
case CB_INITIALISE
case CB_BFORREADNOTES
case CB_BFORWRITECOM
case CB_AFTERWRITECOM
case CB_BFORCOMACTION
case CB_BFORNOTESACTION
case CB_TERMINATE
end select
```

28 Maint [Public Variable]

Version:
Availability:

Maint (Integer)

True if the debugging mode is enabled, otherwise False.

29 MessageDialog [Function]

Version:
Availability:

MessageDialog (Message As String, Icon As Long, Title As String) As Integer

This function is calling the standard Lotus Script messagebox() function. However, it will disable the progress bar for the time the message is displayed and enable it afterwards. The progress bar will not always allow a messagebox to be displayed whilst active and therefore this function disables it to ensure the messagebox() function works properly.

Use this function to display any messages or confirmation dialogs within the script stored in the Callback Script or Error Handler field of the Integra profile.

Parameters and Return Value:

Please refer to the Notes help for the messagebox() function.

Example:

Call MessageDialog("An error has occurred. Please...",16,"Integra Runtime Message")

30 MultiValueFdToArray [Function]

Version:
Availability:

MultiValueFdToArray (Item As Variant, Text() As String) As Integer

This function converts a multi-value form field into an array of strings.

Parameters:

Item (Variant)

Array of type variant containing the multi-values of the Notes document field received with NoteDocument.GetItemValue().

Text (String-Array)

Dynamic declared blank array of strings.

Return Value:

Integer. Number of array elements.

Example:

```
Dim ws As New NotesUiWorkspace
Dim doc As NotesDocument
Dim item As Variant
Dim text() As String

Set doc = ws.CurrentDocument.Document

item = doc.GetItemValue("Categories")

Call MultiValueFdToArray(item,text)
```

31 NotesSortString [Function]

Version:
Availability:

Function NotesSortString (txt As String, descending As Integer) As String

Parameters:

Return Value:

Example:

```
' Callback Script Example
Const CB_INITIALISE = 3
Const CB_BFORREADNOTES = 6
Const CB_BFORWRITECOM = 1
Const CB_AFTERWRITECOM = 9
Const CB_BFORCOMACTION = 5
Const CB_BFORNOTESACTION = 2
Const CB_TERMINATE = 4

select case stats.cbstatus
case CB_INITIALISE
case CB_BFORREADNOTES
case CB_BFORWRITECOM
case CB_AFTERWRITECOM
case CB_BFORCOMACTION
case CB_BFORNOTESACTION
case CB_TERMINATE
end select
```

32 NotesVersionNumber [Function]

Version:
Availability:

NotesVersionNumber ()

This function returns the Notes version as a numeric value

Return Value:

Variant. Integer or floating point.

Example:

```
Dim ver As Variant

ver = NotesVersionNumber()

Call MsgBox(Format(ver),0,"Notes Version Number")
```

33 OpenFileDialog [Function]

Version:
Availability:

OpenFileDialog (MultiSelect As Integer, Message As String, DefaultDir As String) As String

This function displays the Windows File-Open dialog. It disables the progress bar before activating the dialog and enables it afterwards.

Parameters:

MultiSelect (Integer)

True or False. If True, the File-Open dialog will allow the selection of multiple files. In this case OpenFileDialog will return one string of filenames each filename separated with a semicolon. If False, the File-Open dialog allows the selection of a single file only.

Message (String)

A prompt/title displayed in the Open-File-Dialog.

DefaultDir (String)

Default directory the Open-File-Dialog opens with.

Return Value:

String. Comma separated list of files (including their pathname).

Example:

```
Dim filename As String
```

```
Filename = OpenFileDialog(True, "Please select the file(s) to be processed", GetIntegraDataDir())
```

```
Call MessageBox(FileName, 0, "You selected the following files...")
```

34 ProgBar [Public Variable]

Version:

Availability:

ProgBar (Integer)

True if the progress bar is enabled, otherwise False.

35 QuickSort [Procedure]

Version:

Availability:

QuickSort (SortArray() As String, Descending As Integer)

This procedure is using the quick sort algorithm to sort the elements in an array. I can do so either ascending or descending order.

Parameters:

SortArray (Array of Strings)

Array of text string elements.

Descending (Integer)

Integer. True or False. If True QuickSort will sort in descending order.

Example:

```
Dim arr(4) As String  
Dim lst As String
```

```
arr(0) = "bananas"  
arr(1) = "grapes"  
arr(2) = "apples"  
arr(3) = "pears"  
arr(4) = "oranges"
```

```
Call QuickSort(arr, False)
```

```
lst = ArrayToString(arr, ", ")
```

```
Call MessageBox(lst, 0, "Sorted List")
```

36 RemoveCR [Function]

Version:
Availability:

RemoveCR (Text As String) As String

This function removes any Chr(10) or Chr(13) elements from the passed on text string.

Parameters:

Text (String)

Any text string.

Return Value:

String.

Example:

```
Dim ws As New NotesUIWorkspace
Dim doc As NotesDocument
Dim txt As String

Set doc = ws.CurrentDocument.Document

txt = RemoveCR(doc.Description(0))

Call MessageDialog(txt,0,"Remove Carriage Returns")
```

37 ReplaceStr [Function]

Version:
Availability:

ReplaceStr (SearchStr As String, FindStr As String, ReplaceStr As String) As String

This function searches the SearchStr for any occurrence of FindStr and replaces it with ReplaceStr.

Parameters:

SearchStr (String)

The string that is being searched for the occurrence of FindStr.

FindStr (String)

The string that is being searched for in SearchStr.

ReplaceStr (String)

The string that replaces FindStr where ever it occurs in SearchStr.

Return Value:

String.

Example:

```
Dim ws As New NotesUIWorkspace
Dim doc As NotesDocument
Dim newstr As String

Set doc = ws.CurrentDocument.Document

newstr = ReplaceStr(doc.NamesList(0),";",";")

Call MessageDialog(newstr,0,"New String")
```

38 ResponseDocLevel [Function]

Version:
Availability:

ResponseDocLevel (Note As NotesDocument) As Integer

This function returns the response document level of a document. A return value of 0 (zero) indicates that the document is a (top level) parent document; any value larger than zero represents the response level to the top level parent document.

Parameters:

Note (NotesDocument)
Object handle to a Notes back-end Document.

Return Value:

Integer.

Example:

```
Dim ws As New NotesUIWorkspace
Dim doc As NotesDocument
Dim level As Integer

Set doc = ws.CurrentDocument.Document

level = ResponseDocLevel(doc)

Call MessageDialog("Level = " & level,0,"Response Document Level")
```

39 SaveFileDialog [Function]

Version:
Availability:

SaveFileDialog (Message As String, DefaultDir As String) As String

This function displays the Windows File-Save dialog. It disables the progress bar before activating the dialog and enables it afterwards.

Parameters:

Message (String)
A prompt/title displayed in the File-Open dialog.
DefaultDir (String)
Default directory the Save-File-Dialog opens with.

Return Value:

String.

Example:

```
Dim filename As String

Filename = OpenSaveDialog(True, "Please select the filename you want to save the currently processed document with...")
Call MessageDialog(Filename,0,"You entered the following filename...")
```

40 ScheduledRun [Public Variable]

Version:
Availability:

ScheduledRun (Integer)

True if this is a scheduled execution running on the server, otherwise False.

41 ScriptExecute [Function]

Version:
Availability:

Function MyExecute (pbaroff As Integer) As Integer

Parameters:

Return Value:

Example:

```
' Callback Script Example
Const CB_INITIALISE = 3
Const CB_BFORREADNOTES = 6
Const CB_BFORWRITECOM = 1
Const CB_AFTERWRITECOM = 9
Const CB_BFORCOMACTION = 5
Const CB_BFORNOTESACTION = 2
Const CB_TERMINATE = 4

select case stats.cbstatus
case CB_INITIALISE
case CB_BFORREADNOTES
case CB_BFORWRITECOM
case CB_AFTERWRITECOM
case CB_BFORCOMACTION
case CB_BFORNOTESACTION
case CB_TERMINATE
end select
```

42 SendMail [Function]

Version:
Availability:

SendMail (SendTo() As String, CopyTo() As String, Bcc() As String, Subject As String, BodyHdr As String, Filename As Variant, ConfirmDlg As Integer, AppendRtitem As NotesRichTextItem, KillFile As Integer, ShowPBar As Integer) As Integer

This function is creating a new blank memo document in the user's mail database and calls then the CreateMemo() function to populate the memo with the necessary or provided information (parameters) to make it a valid (send able) memo document.

In case this function is called in a scheduled profile or from web-browser initiated profile, then the memo document is created in the Integra database.

Note: the memo document is not saved to file and only temporarily created as a container for the to be sent memo.

Parameters:

SendTo (Array of Strings)

List of people (or groups) that will receive the memo.

CopyTo (Array of Strings)

List of people (or groups) that will receive a copy of the memo.

Bcc (Array of Strings)

List of people (or groups) that will receive a blind copy of the memo.

Subject (String)

Subject for the memo.

BodyHdr (String)

Heading text for the memo's body part. The provided text will be placed before any attachment.

Filename (Variant)

Single filename (String) or list of filenames (Array of Strings) of a file or files to be attached to the memo document. Filenames have to include the file path.

ConfirmDlg (Integer)

True or False. If True the user will be presented with a mail-send dialog allowing to enter the mail recipients, subject and body heading for the memo document.

AppendRtitem (NotesRichtextItem)

NotesRichtextItem object which rich text contents will be appended to the body field of the memo document.

KillFile (Integer)

True or False. If True the file passed in the parameter Filename will be deleted (from disk) after being attached to the memo document.

ShowPBar (Integer)

True or False. If True, a progress bar will be displayed whilst attaching the file(s).

Return Value:

Integer. True or False.

Example:

```
Dim sendto() As String
Dim cc() As String
Dim bcc() As String
```

```
redim sendto(0)
redim cc(0)
redim bcc(0)
```

```
sendto(0) = "All Staff"
cc(0) = ""
bcc(0) = ""
```

```
If Not SendMail(sendto,cc,bcc,"SendMail Function","This is a body heading text","", False,Nothing,False,False) Then
```

```
    Call MessageDialog("An error has occurred while sending the mail", 0, "Send Mail")
End If
```

43 SendMailAttachment [Function]

Version:

Availability:

SendMailAttachment (SendTo() As String, Subject As String, Filename As Variant, ConfirmDlg As Integer, AppendRtitem As NotesRichtextItem, KillFile As Integer, ShowPBar As Integer) As Integer

This function is not more in use and is kept only for compatibility reasons to Integra 2.2. Please use the SendMail() function.

See SendMail() function.

44 SetUserProfileVar [Procedure]

Version:

Availability:

SetUserProfileVar (VarName As String, Value As Variant)

The very first time a user is running an Integra import or export a personal user profile is created in the Integra database to store temporary runtime information.

The user profile can be utilised to temporarily store variables between the different event calls of the Callback Script. This procedure is saving any value that can be stored in a Notes document field temporarily in the user's personal profile and can be retrieved back using the GetUserProfileVar() function.

Parameters:

VarName (String)

Name of the variable (field) under which the contents of Value is to be saved in the user's personal profile.

Value (Variant)

Any value of the following types: String, Numeric, Boolean, DateTime or an Array of these data types.

Example:

See GetUserProfileVar()

45 StringToArray [Function]

Version:

Availability:

StringToArray (Item As Variant, Separator As String, StrArray() As String) As Integer

This function is similar to the Notes @Explode() function. It converts Item into array elements if they are separated by the specified Separator. If Item is not of type String Item will be converted to String.

Parameters:

Item (Variant)

Text string, date-time or numeric value. Item will be converted to String before converting it to array elements.

Separator (String)

One or more characters in Item that are used to define the end of an element.

StrArray (Array of Strings)

Blank dynamic declared array of strings in which StringToArray() will return the separated string elements.

Return Value:

Integer. Upper bound of StrArray

Example:

```
Dim arr() As String
Dim lst As String
Dim num As String
```

```
lst = "apples,oranges,bananas,pears,grapes"
```

```
num = StringToArray(lst,",",arr)
```

```
Call MsgBox("Number of elements: " & Format(num),0,"String To Array")
```

46 TopParent [Function]

Version:

Availability:

Function TopParent (doc As notesdocument, lst As String, sep As String) As notesdocument

Parameters:

Return Value:

Example:

```
' Callback Script Example
Const CB_INITIALISE = 3
Const CB_BFORREADNOTES = 6
Const CB_BFORWRITECOM = 1
Const CB_AFTERWRITECOM = 9
Const CB_BFORCOMACTION = 5
Const CB_BFORNOTESACTION = 2
```

```
Const CB_TERMINATE = 4
```

```
select case stats.cbstatus  
case CB_INITIALISE  
case CB_BFORREADNOTES  
case CB_BFORWRITECOM  
case CB_AFTERWRITECOM  
case CB_BFORCOMACTION  
case CB_BFORNOTESACTION  
case CB_TERMINATE  
end select
```

47 WebRun [Public Variable]

Version:

Availability:

WebRun (Integer)

True if this is a web-browser initiated execution running on the server, otherwise False.