SMTPit Pro

- SMTPit_AddEmailHeader
- SMTPit_Clear
- SMTPit_Configure
- SMTPit_Connect
- SMTPit_Disconnect
- SMTPit_File_Copy
- SMTPit_File_Delete
- SMTPit_File_Exists
- SMTPit_File_Export
- SMTPit_File_GetPath
- SMTPit_File_Move
- SMTPit_File_Rename
- SMTPit_File_SelectFileWithDialog
- SMTPit_File_SelectFolderWithDialog
- SMTPit_File_Size
- SMTPit_GetSendValue
- SMTPit_IsConnected
- SMTPit_MergeText
- SMTPit_QuickSet
- SMTPit_Register
- SMTPit_Send
- SMTPit_SendWithDialog
- SMTPit_SetAttachment
- SMTPit_SetAttachmentWithDialog
- SMTPit_SetAuthentication
- SMTPit_SetBCC
- SMTPit_SetBody
- SMTPit_SetBodyFooter
- SMTPit_SetBodyHeader
- SMTPit_SetBodyWithFile
- SMTPit_SetCC
- SMTPit_SetCharSet
- SMTPit_SetFrom
- SMTPit_SetHost
- SMTPit_SetInlineImage
- SMTPit_SetInlineImageWithDialog
- SMTPit_SetPriority
- SMTPit_SetReplyTo
- SMTPit_SetSubject
- SMTPit_SetTo
- SMTPit_StatusWindow
- SMTPit_StyledTextToHTML
- SMTPit_VerifyEmailAddress
- SMTPit_Version
- SMTPit_VersionAutoUpdate

SMTPit_AddEmailHeader

Description

The SMTPit_AddEmailHeader function adds extra email headers to your email message. The "Headers" parameter takes a name and value pair string separated by an equal sign. Note, the name and value pair is separated with an equal sign in this function, but when viewed in the recipient's email, it will be shown as separated by a colon. Multiple headers can be added by separating name and value pairs with a return (¶). There should be no spaces in the name section, however the value can have spaces.

Note: You can also set Default Email Headers in the Configuration Dialog. If you have Email Headers assigned in the Configuration Dialog, they will be appended to the Header of the email before any Email Headers you set with this function. You cannot set any of the following with this function: To, From, Subject, Date, Message-ID, CC, or BCC.

Return Type

Text

Format

SMTPit_AddEmailHeader (**Header(s)**)

Required Parameters

Header(s)
The Email Header(s) you want to add.

Examples

Example 1

Code:

```
SMTPit_AddEmailHeader( "X-Company=Your Company Name" )
```

Result:

Adds an X-header named "X-Company" with the value of "Your Company Name"

Example 2

Code:

```
SMTPit_AddEmailHeader( "X-RecID=" & SomeTable::ID & "¶X-URL=http://www.myurl.com" )
```

Result:

Adds two X-headers, one named "X-RecID" with the value in the ID field, and a second named "X-URL" with the value "http://www.myurl.com".

SMTPit_Clear

Description

The SMTPit_Clear function clears all values that have been set with the External Functions. This allows you to send a new message without any lingering values.

However, you can also clear only specific values and thus specifically use values from the last message.

Note: You can also set SMTPit Pro to clear specific fields on the "After Send" tab under the "Extras" tab in the Configuration Dialog.

Return Type

Text

Format

SMTPit_Clear (**Items**)

Optional Parameters

Items
A Paragraph Mark- or Return-Separated list of items to Clear. (Leave off this parameter to clear everything.) Values you can use for this parameter include: "Host", "Port", "Authentication", "Tos", "CCs", "BCCs", "From", "Reply-To", "Subject", "Priority", "Email Headers", "Text Body", "Text Header", "Text Footer", "HTML Body", "HTML Header", "HTML Footer", "HTML Images", and "Attachments". Additionally, you can use "Connection-Related" to clear all connection-related values, "Recipients" to clear all recipient values, "Text" to clear all body text values, and "HTML" to clear all HTML-related values.

Examples

Example 1

Code:

```
SMTPit_Clear
```

Result:

Clears all values used in the last email message.

Example 2

Code:

```
SMTPit_Clear( "Tos" )
```

Result:

Clears the "To" email addresses from the last email message.

Example 3

Code:

```
SMTPit_Clear( "To¶Subject" )
```

Result:

Clears the "To" and "Subject" values from the last email message.

SMTPit_Configure

Description

Calling this function with no parameters will open the SMTPit Pro Configuration Dialog. You can optionally open to a specific tab by specifying the name of the tab as the "Option" parameter. This function also allows you to get or set any preference found in the Configuration Dialog.

Valid PrefNames:

"AddHelp" - If True, Function specific Help will be added to Calculations when inserting the Plug-in's Functions.
"Host" - The Domain Name or IP Address of the Default SMTP Mail Server.
"From" - The Default From Email Address.
"ReplyTo" - The Default Reply-To Email Address. (Leave blank for no Reply-To.)
"Subject" - The Default Email Subject.
"Priority" - The Default Email Priority. (Default Priority is 'Normal'.)
"Timeout" - The Number of Seconds to wait for the Mail Server to respond to a Connection Attempt. (Default Timeout is 30 seconds.)
"LineBreak" - The Number of Characters per line in an Encoded Email. (Default Line Break Column is 80.)
"TextHeader" - Default Text Body Header.
"TextFooter" - Default Text Body Footer.
"HTMLHeader" - Default HTML Body Header.
"HTMLFooter" - Default HTML Body Footer.
"Attachments" - Default Email Attachments.
"HTMLImages" - Default Email Inline Images.
"EmailHeaders" - Default Extra Email Headers.
"ContentTypes" - Attachment and Inline Image Content-Types.
"AuthType" - The Type of Authentication your mail server requires. (Specify "Plain", "Login", "CRAM-MD5", or "Auto".)
"Username" - The Username to use when connecting to your mail server.
"Password" - The Password to use when connecting to your mail server.
"TLSType" - The type of TLS/SSL your mail server requires. (Specify "None", "AfterConnect", or "BeforeConnect".)
"PopupSent" - If True, SMTPit Pro will popup an "Email Sent Successfully." or Error Dialog after sending an email.
"ClearAll" - If True, SMTPit Pro will clear all the items in the list of items to clear from the Configuration Dialog.
"Clears" - A list of items to clear after sending an email.
"CharSet" - The character set for the email. (See the associated setting in the Configuration Dialog for possible values.
"BreakOnBadRcpt" - If True, SMTPit Pro will abort sending an email if any recipient is deemed invalid by the mail server.
"Port" - The TCP/IP Port for your mail server.
"UseOldFunctions" - If True, the old-style External("SMTP-XXX") functions will be available.
"UseNewFunctions" - If True, the new-style SMTPit_XXX functions will be available.

Return Type

Varies

Format

SMTPit_Configure (**Option** ; **PrefName** ; **PrefValue**)

Optional Parameters

Option

Specify the name of a Tab in the Configuration Dialog to show it opened to that tab.
Specify "Get" with the PrefName parameter to get a preference value. (If not found, and PrefValue is defined, PrefValue will be returned.)
Specify "Set" with the PrefName and PrefValue parameters to set a preference value.

PrefName

The Name of the Preference to Get or Set. (See the Function Description for a list of valid PrefNames.)

PrefValue

The Value of the Preference to Set. (See the Function Description for some possible values.)

Examples

Example 1

Code:

```
SMTPit_Configure
```

Result:

Opens the SMTPit Pro Configuration Dialog. (Because the "Option" parameter is not used, the dialog will open to the "Basics" tab.)

Example 2

Code:

```
SMTPit_Configure( "About" )
```

Result:

Opens the SMTPit Pro Configuration Dialog to the "About" tab.

Example 3

Code:

```
SMTPit_Configure( "Get" ; "Host" )
```

Result:

Returns the value of the 'Host' setting from the Configuration Dialog.

Example 4

Code:

```
SMTPit_Configure( "Set" ; "Subject" ; "My Subject" )
```

Result:

Sets the 'Subject' setting in the Configuration Dialog to "My Subject".

SMTPit_Connect

Description

You can use the "SMTPit_Connect" function to specifically connect to your mail server rather than letting the "SMTPit_Send" function connect for you.

When you normally use the "SMTPit_Send" function, SMTPit Pro will connect to your mail server, send your email, and then disconnect. However, if you call the "SMTPit_Connect" function prior to calling the "SMTPit_Send" function, SMTPit Pro will simply connect to your email server and send the message without disconnecting from your server.

This is useful in mass-email solutions that need to send out several email messages at a time. Connecting once, sending several emails, and then disconnecting is much more efficient than connecting and disconnecting for each individual email. If you call "SMTPit_Connect", you must call "SMTPit_Disconnect" after sending all of your messages, otherwise SMTPit will never disconnect from the mail server.

In a mass-email solution, you would want to call SMTPit_Connect before your looping script, and call SMTPit_Disconnect after your looping script. For normal operation, use the empty string as the parameter ("").

If you would like a transcript of the interaction between SMTPit and your mail server, use the optional "GetTranscript" parameter. This may be useful when determining why your mail server is not allowing you to connect to it.

If your mail server requires authentication before you can send email, you must set up the Authentication Type, Username, and Password before calling this function. Otherwise, SMTPit Pro will not have the required information to properly authenticate with your mail server.

Return Type

Text

Format

SMTPit_Connect (**GetTranscript**)

Optional Parameters

GetTranscript
Set to True if you want a Transcript of the interaction between SMTPit and your Mail Server.

Related Items

SMTPit_Disconnect, SMTPit_Send

Examples

Example 1

Code:

```
SMTPit_Connect
```

Result:

Connects to your mail server

Example 2

Code:

```
SMTPit_Connect( "True")
```

Result:

Connects to your mail server and returns a transcript.

SMTPit_Disconnect

Description

If you use SMTPit_Connect to connect to your mail server manually (instead of letting SMTPit_Send do it), then you must call SMTPit_Disconnect to disconnect from your mail server. Otherwise, SMTPit will never disconnect you from the mail server.

Return Type

Text

Format

SMTPit_Disconnect (**GetTranscript**)

Optional Parameters

GetTranscript
Set to True if you want a Transcript of the interaction between SMTPit and your Mail Server.

Related Items

SMTPit_Connect, SMTPit_Send

Examples

Example 1

Code:

```
SMTPit_Disconnect
```

Result:

Disconnects from your mail server

Example 2

Code:

```
SMTPit_Disconnect( GetTranscript )
```

Result:

Disconnects from your mail server and returns a transcript.

SMTPit_File_Copy

Description

You can use this function to copy a file to a new location.

Return Type

Text

Format

SMTPit_File_Copy (**Source** ; **Destination**)

Required Parameters

Source
The Path and File Name of the File you want to Copy.

Destination
The Path to the Folder where you want the Copy. (You can optionally specify a File Name to Rename the File.)

Related Items

SMTPit_File_Delete, SMTPit_File_Exists, SMTPit_File_Export, SMTPit_File_GetPath, SMTPit_File_Move, SMTPit_File_Rename, SMTPit_File_SelectFileWithDialog, SMTPit_File_SelectFolderWithDialog, SMTPit_File_Size

Examples

Example 1

Code:

```
SMTPit_File_Copy( "/source/copy.txt" ; "/destination/" )
```

Result:

Copies the file "copy.txt" from the "source" folder to the "destination" folder on Mac.

Example 2

Code:

```
SMTPit_File_Copy( "c:\source\copy.txt" ; "c:\destination\ " )
```

Result:

Copies the file "copy.txt" from the "source" folder to the "destination" folder on Windows.

Example 3

Code:

```
SMTPit_File_Copy( "c:\source\copy.txt" ; "c:\destination\copied.txt" )
```

Result:

Copies the file "copy.txt" from the "source" folder to the "destination" folder and renames the file "copied.txt" on Windows.

Example 4

Code:

```
SMTPit_File_Copy( "/source/copy.txt" ; "/destination/copied.txt" )
```

Result:

Copies the file "copy.txt" from the "source" folder to the "destination" folder and renames the file "copied.txt" on Mac.

Example 5

Code:

```
SMTPit_File_Copy( "/Users/John/Documents/list.txt" ; "/Volumes/FileServer/list.txt" )
```

Result:

Copies the file "list.txt" from the "Documents" folder to the "FileServer" network share on Mac. The network share must be mounted prior to calling this function, as it is unable to mount a share.

SMTPit_File_Delete

Description

You can use this function to delete a file.

Return Type

Text

Format

SMTPit_File_Delete (**File**)

Required Parameters

File
The Path and File Name of the File you want to Delete.

Related Items

SMTPit_File_Copy, SMTPit_File_Exists, SMTPit_File_Export, SMTPit_File_GetPath, SMTPit_File_Move, SMTPit_File_Rename, SMTPit_File_SelectFileWithDialog, SMTPit_File_SelectFolderWithDialog, SMTPit_File_Size

Examples

Example 1

Code:

```
SMTPit_File_Delete( "c:\destination\deleteme.txt" )
```

Result:

Deletes the file "deleteme.txt" on Windows.

Example 2

Code:

```
SMTPit_File_Delete( "/Images/deleteme.txt" )
```

Result:

Deletes the file "deleteme.txt" on Mac.

SMTPit_File_Exists

Description

This function will check to see if a file exists.

Return Type

Number (1=True, 0=False)

Format

SMTPit_File_Exists (**File**)

Required Parameters

File
The Path and File Name of the File you want to Find.

Related Items

SMTPit_File_Copy, SMTPit_File_Delete, SMTPit_File_Export, SMTPit_File_GetPath, SMTPit_File_Move, SMTPit_File_Rename, SMTPit_File_SelectFileWithDialog, SMTPit_File_SelectFolderWithDialog, SMTPit_File_Size

Examples

Example 1

Code:

```
SMTPit_File_Exists( "c:\random\ithink.txt" )
```

Result:

Returns "True" if the "ithink.txt" file exists on Windows.

Example 2

Code:

```
SMTPit_File_Exists( "/random/ithink.txt" )
```

Result:

Returns "True" if the "ithink.txt" file exists on Mac.

SMTPit_File_Export

Description

This function will export a file from a container to a specified path. SMTPit Pro will use the name of the file in the container unless you specify a name with the "SaveAs" parameter.

Return Type

Text

Format

SMTPit_File_Export (**Container** ; **Path** ; **SaveAs**)

Required Parameters

Container
The Container that contains the File you are Exporting.

Path
The Path to the Folder where you want the Exported File.

Optional Parameters

SaveAs
The File Name you want the Exported File to have.

Related Items

SMTPit_File_Copy, SMTPit_File_Delete, SMTPit_File_Exists, SMTPit_File_GetPath, SMTPit_File_Move, SMTPit_File_Rename, SMTPit_File_SelectFileWithDialog, SMTPit_File_SelectFolderWithDialog, SMTPit_File_Size

Examples

Example 1

Code:

```
SMTPit_File_Export( Container ; "c:\exports" )
```

Result:

Exports the file in the "Container" field to "c:\exports" on Windows.

Example 2

Code:

```
SMTPit_File_Export( Container ; "/exports/" )
```

Result:

Exports the file in the "Container" field to "/exports/" on Mac.

Example 3

Code:

```
SMTPit_File_Export( Container ; "c:\exports" ; "newexport.txt" )
```

Result:

Exports the file in the "Container" field to "c:\exports\newexport.txt".

Example 4

Code:

```
SMTPit_File_Export( Container ; "/exports/" ; "newexport.txt" )
```

Result:

Exports the file in the "Container" field to "/exports/newexport.txt".

SMTPit_File_GetPath

Description

You can use the SMTPit_File_GetPath to retrieve a variety of paths including: Database, FileMaker, Root, System, Desktop, Preferences, Temporary, Applications, or Documents.

This function can be used on concert with other functions. For example you could use SMTPit_File_GetPath to get the user desktop path and then use SMTPit_File_Export to export a file in a container field.

Return Type

Text

Format

SMTPit_File_GetPath (**Type**)

Required Parameters

Type
Which Path Type you need. Specify "Database", "FileMaker", "Root", "System", "Desktop", "Preferences", "Temporary", "Applications", or "Documents".

Related Items

SMTPit_File_Copy, SMTPit_File_Delete, SMTPit_File_Exists, SMTPit_File_Export, SMTPit_File_Move, SMTPit_File_Rename, SMTPit_File_SelectFileWithDialog, SMTPit_File_SelectFolderWithDialog, SMTPit_File_Size

Examples

Example 1

Code:

```
SMTPit_File_GetPath( "Database" )
```

Result:

Returns the path to the current database.

Example 2

Code:

```
SMTPit_File_GetPath( "Desktop" )
```

Result:

Returns the path to desktop.

SMTPit_File_Move

Description

You can use this function to move a file to a new location.

Return Type

Text

Format

SMTPit_File_Move (**Source** ; **Destination**)

Required Parameters

Source
The Path and File Name of the File you want to Move.

Destination
The Path to the Folder where you want the File. (You can optionally specify a File Name to Rename the File.)

Related Items

SMTPit_File_Copy, SMTPit_File_Delete, SMTPit_File_Exists, SMTPit_File_Export, SMTPit_File_GetPath, SMTPit_File_Rename, SMTPit_File_SelectFileWithDialog, SMTPit_File_SelectFolderWithDialog, SMTPit_File_Size

Examples

Example 1

Code:

```
SMTPit_File_Move( "/source/move.txt" ; "/destination/" )
```

Result:

Moves the file "move.txt" from the "source" folder to the "destination" folder on Mac.

Example 2

Code:

```
SMTPit_File_Move( "c:\source\move.txt" ; "c:\destination\ " )
```

Result:

Moves the file "move.txt" from the "source" folder to the "destination" folder on Windows.

Example 3

Code:

```
SMTPit_File_Move( "c:\source\move.txt" ; "c:\destination\moved.txt" )
```

Result:

Moves the file "move.txt" from the "source" folder to the "destination" folder and renames the file "moved.txt" on Windows.

Example 4

Code:

```
SMTPit_File_Move( "/source/move.txt" ; "/destination/moved.txt" )
```

Result:

Moves the file "move.txt" from the "source" folder to the "destination" folder and renames the file "moved.txt" on Mac.

SMTPit_File_Rename

Description

You can use this function to rename a file.

Return Type

Text

Format

SMTPit_File_Rename (**File** ; **NewName**)

Required Parameters

File
The Path and File Name of the File you want to Rename.

NewName
The New Name for the File.

Related Items

SMTPit_File_Copy, SMTPit_File_Delete, SMTPit_File_Exists, SMTPit_File_Export, SMTPit_File_GetPath, SMTPit_File_Move, SMTPit_File_SelectFileWithDialog, SMTPit_File_SelectFolderWithDialog, SMTPit_File_Size

Examples

Example 1

Code:

```
SMTPit_File_Rename( "/Images/rename.jpg" ; "renamed.jpg" )
```

Result:

Renames the file "rename.jpg" to "renamed.jpg" on Mac.

Example 2

Code:

```
SMTPit_File_Rename( "c:\images\rename.jpg" ; "renamed.jpg" )
```

Result:

Renames the file "rename.jpg" to "renamed.jpg" on Windows.

SMTPit_File_SelectFileWithDialog

Description

This function will display a file chooser dialog. You can set the location to open the dialog to, as well as display a custom prompt message.

Return Type

Text

Format

SMTPit_File_SelectFileWithDialog (**StartPath** ; **Prompt**)

Optional Parameters

StartPath
The Path to the Folder to display when the dialog first opens.

Prompt
An optional Prompt or Message to display on the dialog.

Related Items

SMTPit_File_Copy, SMTPit_File_Delete, SMTPit_File_Exists, SMTPit_File_Export, SMTPit_File_GetPath, SMTPit_File_Move, SMTPit_File_Rename, SMTPit_File_SelectFolderWithDialog, SMTPit_File_Size

Examples

Example 1

Code:

```
SMTPit_File_SelectFileWithDialog
```

Result:

Displays a file chooser dialog, and returns the path of the chosen file.

Example 2

Code:

```
SMTPit_File_SelectFileWithDialog( "c:\images" )
```

Result:

Displays a file chooser dialog starting at "images" directory on Windows.

Example 3

Code:

```
SMTPit_File_SelectFileWithDialog( "/Images/" )
```

Result:

Displays a file chooser dialog starting at "images" folder on Mac.

Example 4

Code:

```
SMTPit_File_SelectFileWithDialog( "" ; "Choose a file..." )
```

Result:

Displays a file chooser with the prompt "Choose a file...".

SMTPit_File_SelectFolderWithDialog

Description

This function will display a folder chooser dialog. You can set the location to open the dialog to, as well as display a custom prompt message.

Return Type

Text

Format

SMTPit_File_SelectFolderWithDialog (**StartPath** ; **Prompt**)

Optional Parameters

StartPath
The Path to the Folder to display when the dialog first opens.

Prompt
An optional Prompt or Message to display on the Dialog.

Related Items

SMTPit_File_Copy, SMTPit_File_Delete, SMTPit_File_Exists, SMTPit_File_Export, SMTPit_File_GetPath, SMTPit_File_Move, SMTPit_File_Rename, SMTPit_File_SelectFileWithDialog, SMTPit_File_Size

Examples

Example 1

Code:

```
SMTPit_File_SelectFolderWithDialog
```

Result:

Displays a folder chooser dialog, and returns the path of the chosen folder.

Example 2

Code:

```
SMTPit_File_SelectFolderWithDialog( "c:\images" )
```

Result:

Displays a folder chooser dialog starting at "images" directory on Windows.

Example 3

Code:

```
SMTPit_File_SelectFolderWithDialog( "/Images/" )
```

Result:

Displays a folder chooser dialog starting at "images" folder on Mac.

Example 4

Code:

```
SMTPit_File_SelectFolderWithDialog( "" ; "Choose a folder..." )
```

Result:

Displays a folder chooser with the prompt "Choose a folder...".

SMTPit_File_Size

Description

This function returns the size of a file.

Return Type

Number

Format

SMTPit_File_Size (**File**)

Required Parameters

File
The Path and File Name of the File you need the size of.

Related Items

SMTPit_File_Copy, SMTPit_File_Delete, SMTPit_File_Exists, SMTPit_File_Export, SMTPit_File_GetPath, SMTPit_File_Move, SMTPit_File_Rename, SMTPit_File_SelectFileWithDialog, SMTPit_File_SelectFolderWithDialog

Examples

Example 1

Code:

```
SMTPit_File_Size( "c:\images\big.jpg" )
```

Result:

Returns the size of the "big.jpg" file on Windows.

Example 2

Code:

```
SMTPit_File_Size( "/Images/big.jpg" )
```

Result:

Returns the size of the "big.jpg" file on Mac.

SMTPit_GetSendValue

Description

This function retrieves values entered in the last Email Window or last email sent. It can be used for archiving the sent email. You can also use this function in concert with the "SMTPit_Send" function to send a message after the Email Window has been closed.

Return Type

Text

Format

SMTPit_GetSendValue (**Which**)

Required Parameters

Which
Which Send Value to retrieve. (Specify "To", "CC", "BCC", "From", "Subject", "TextBody", "HTMLBody", "Attachment1", "Attachment2", "Attachment3", "Attachments", "InlineImages", "Reply-To", "Priority", or "Message-ID".)

Related Items

SMTPit_Send, SMTPit_SendWithDialog

Examples

Example 1

Code:

```
SMTPit_GetSendValue( "Subject" )
```

Result:

Returns something like "Test Message Subject"

Example 2

Code:

```
SMTPit_GetSendValue( "To" )
```

Result:

Returns something like "Joe Cool <Joe@cool.com>"

SMTPit_IsConnected

Description

This function will tell you whether or not you are currently connected to a mail server.

Return Type

Number (1=True, 0=False)

Format

SMTPit_IsConnected

Related Items

SMTPit_Connect, SMTPit_Disconnect

Example

Code:

```
SMTPit_IsConnected
```

Result:

Returns 1 if you are currently connected to a mail server.

SMTPit_MergeText

Description

With SMTPit Pro, you can use Merge Fields in your email messages. A merge field is a tag representing a field which will get replaced with the field's contents when the message is sent. For example, if you have a field called "Name", you can represent this field in your email message as "<<Name>>". If your "Name" field happens to be in a table called "Contacts", then you would use "<<Contacts::Name>>" as your merge field tag. If the value in the "Name" field is "Joe Cool", then SMTPit Pro will replace "<<Name>>" with the text "Joe Cool" when the email is sent.

To help you preview what an email message might look like after it is merged, we have included the "SMTPit_MergeText" function. Remember, this function is only for previewing, there is no need to include it in your send mail calculations.

Merge fields can be used in the subject, header, body, and footer of your email messages. SMTPit Pro will automatically replace merge fields with their corresponding field values when sending a message. If SMTPit Pro cannot find the field you are trying to reference, it will place a question mark "?" in place of the merge field.

Return Type

Text

Format

SMTPit_MergeText (**Text**)

Required Parameters

Text
Your Text that contains Merge Fields.

Example

Code:

```
SMTPit_MergeText( Body )
```

Result:

Returns the "Body" field text with all merge fields replaced with their corresponding values.

SMTPit_QuickSet

Description

The SMTPit_QuickSet function allows you to set the From, To, Subject, and TextBody in a quick an simple format. You can also use functions for each of these items.

Return Type

Text

Format

SMTPit_QuickSet (**From** ; **To** ; **Subject** ; **TextBody**)

Required Parameters

From
The From Email Address.

To
The To Email Address.

Subject
The Email Subject.

TextBody
The Email Content.

Related Items

SMTPit_SetBody, SMTPit_SetFrom, SMTPit_SetSubject, SMTPit_SetTo

Example

Code:

```
SMTPit_QuickSet( "me@my.com" ; "you@your.com" ; "Test Message" ; "Test body" )
```

Result:

Sets the from, to, subject, and text body as above.

SMTPit_Register

Description

You can use this function to Register the plug-in from a script instead of through the Configuration Dialog. This is useful when the plug-in is being distributed to many computers, allowing you to intall and register the plug-in without having to physically visit each computer. This function also allows you to check if the plug-in is already registered or clear the current registration. The plug-in always requires you to accept the License Agreement to use the plug-in. This is usually done by presenting the License Agreement Dialog, but that can be suppressed by using the special option value "I Accept the License Agreement".

Return Type

Text

Format

SMTPit_Register (**FirstName** ; **LastName** ; **LicenseKey** ; **Option**)

Required Parameters

FirstName
The Registration First Name you specified when you ordered. (See your Receipt.)

LastName
The Registration Last Name you specified when you ordered. (See your Receipt.)

LicenseKey
The License Key from your Receipt.

Optional Parameters

Option
Specify "Dialog" to enter your Registration Information in a dialog.
Specify "Check" to see if the plug-in is already Registered.
Specify "Clear" to remove the Registration.
Specify "I Accept the License Agreement" to automatically accept the License Agreement dialog without showing it to the end user.
Notes: The "Dialog", "Check", and "Clear" options can also be specified as the first parameter. If you have a Developer License, do not use the "I Accept the License Agreement" value here. See your Developer Instructions file for more information.

Examples

Example 1

Code:

```
SMTPit_Register( "My First Name" ; "My Last Name" ; "My License Key" )
```

Result:

Registers the plug-in with the provided registration information (obviously the above is not valid registration information; please see your Receipt).

Example 2

Code:

```
SMTPit_Register( "Dialog" )
```

Result:

Displays a dialog for you to enter your First Name, Last Name, and License Key as it appears on your Receipt.

Example 3

Code:

```
SMTPit_Register( "Check" )
```

Result:

Returns "Not Registered." or "Registered to <name> for a <license>."

Example 4

Code:

```
SMTPit_Register( "My Company Name" ; "My Company Name" ; "My Site License Key" ; "I Accept
```

Result:

Registers the plug-in and uses the "I Accept the License Agreement" option to keep the License Agreement dialog from appearing.

SMTPit_Send

Description

The SMTPit_Send function first checks to make sure you have assigned enough fields to send an email. If everything checks out, it attempts to send the email to your mail server. If there is a problem sending the email to your mail server, it will return the error. If the email sent successfully, it will return the string "Email Sent Successfully."

You can use this function by itself, or with the SMTPit_Connect and SMTPit_Disconnect functions. If you use this function by itself, this function will connect to your mail server, send the email, and then disconnect. If you call SMTPit_Connect before calling this function, this function will simply send the email; it will not try to connect and disconnect. This is useful in mass-email solutions that send out several emails in one setting. Connecting once, sending several emails, and then disconnecting is much more efficient than connecting and disconnecting for each individual email. If you call SMTPit_Connect, you must call SMTPit_Disconnect, otherwise SMTPit will never disconnect you from the mail server.

If your mail server requires authentication before you can send email, you must set up the Authentication Type, Username, and Password before calling this function. Otherwise, SMTPit will not have the required information to properly authenticate with your mail server.

Return Type

Text

Format

SMTPit_Send (**GetTranscript** ; **ShowResultDialog**)

Optional Parameters

GetTranscript
Set to True if you want a Transcript of the interaction between SMTPit and your Mail Server.

ShowResultDialog
Set to True if you want a Dialog to Popup with the Success or Failure of the Send.

Related Items

SMTPit_Connect, SMTPit_Disconnect, SMTPit_SendWithDialog

Examples

Example 1

Code:

```
SMTPit_Send
```

Result:

Sends an email message based on the settings you have used in other functions and/or in the preferences.

Example 2

Code:

```
SMTPit_Send( True )
```

Result:

Send the message and also returns a transcript.

Example 3

Code:

```
SMTPit_Send( False ; True )
```

Result:

Sends the message and displays a dialog with the result.

Example 4

Code:

```
SMTPit_Send( True ; True )
```

Result:

Sends the message, returns a transcript, and displays a dialog with the result.

SMTPit_SendWithDialog

Description

This function presents the user with a dialog for composing an email. There is both a Simple and Advanced version of the dialog. (The Simple version looks like the Message Dialog in MMEmail). If you specify True for the SendEmail parameter, then when the user presses the "Send" button on the dialog, SMTPit Pro will send the email.

Return Type

Text

Format

SMTPit_SendWithDialog (**Type** ; **AddressList** ; **MergeFields** ; **Containers** ; **SendEmail** ; **GetTranscript** ; **ShowResultDialog**)

Optional Parameters

Type
The Dialog Type you want to use. Specify "" or "Advanced" for the Advanced Dialog or "Simple" for the Simple Dialog (from MMEmail).

AddressList
A Paragraph Mark- or Return-Separated list of Email Addresses for the user to choose from for the To, CC, and BCC fields. Submenus can be added to the menus by placing a greater than sign ">" between menu items.

MergeFields
A Paragraph Mark- or Return-Separated list of Merge Fields the user can Insert from the Contextual Menu.

Containers
A Paragraph Mark- or Return-Separated list of Containers the user can specify as the source for Attachments.

SendEmail
If True, the dialog will actually Send the Message when the user presses "Send". If False, the Dialog will simply close, allowing you to manipulate the data before sending. (Default is True.)

GetTranscript
Set to True if you want a Transcript of the interaction between SMTPit and your Mail Server.

ShowResultDialog
Set to True if you want a Dialog to Popup with the Success or Failure of the Send.

Related Items

SMTPit_Connect, SMTPit_Disconnect, SMTPit_Send

SMTPit_SetAttachment

Description

The SMTPit_SetAttachment function allows you to attach a file from your hard drive or network to the current email being sent. You must specify the full path and filename of the file. You can set more than one file with this function by separating the file names with the paragraph mark or, if in a field, with a return. You can also simply reference a container field to attach the file in the container field.

Return Type

Text

Format

SMTPit_SetAttachment (**Attachment(s)** ; **SMTPit_Append**)

Required Parameters

Attachment(s)
Specify the Path(s) and File Name(s) of the Attachment(s) on the Hard Drive (separate multiple Attachments with a Paragraph Mark or Return); or specify a Container Field that holds the Attachment.

Optional Parameters

SMTPit_Append
Leave this in to add the(se) Attachment(s) to the current list of Attachments.

Related Items

SMTPit_SetAttachmentWithDialog

Examples

Example 1

Code:

```
SMTPit_SetAttachment( "c:\My Documents\resume.doc")
```

Result:

Attaches the file "resume.doc" on windows.

Example 2

Code:

```
SMTPit_SetAttachment( "/Documents/resume.doc")
```

Result:

Attaches the file "resume.doc" on Mac.

Example 3

Code:

```
SMTPit_SetAttachment( "c:\pdfs\Manual.pdf¶c:\pdfs\How to Install.pdf" )
```

Result:

Attaches the "Manual.pdf" and "How to Install.pdf" files on Windows.

Example 4

Code:

```
SMTPit_SetAttachment( MyEmailAttachmentsField)
```

Result:

Attaches the file in the MyEmailAttachmentsField. (The field can either contain a path, or be a container field with a file in it.)

Example 5

Code:

```
SMTPit_SetAttachment( Container1 ) & ¶ &
SMTPit_SetAttachment( Container2 ; SMTPit_Append )
```

Result:

Attaches the files in the Container1 and the Container2 fields. When calling this function multiple times, the second (and further) function calls must include the 2nd parameter of SMTPit_Append, otherwise the second function call would overwrite the data set by the first call.

SMTPit_SetAttachmentWithDialog

Description

This function presents you with a standard open file dialog that lets you choose a file from your hard drive. It then attaches the chosen file to the current email.

Return Type

Text

Format

SMTPit_SetAttachmentWithDialog (**SMTPit_Append**)

Optional Parameters

SMTPit_Append
Leave this in to add the Attachment to the current list of Attachments

Related Items

SMTPit_SetAttachment

Examples

Example 1

Code:

```
SMTPit_SetAttachmentWithDialog
```

Result:

Attaches the file you choose from an open file dialog.

Example 2

Code:

```
SMTPit_SetAttachmentWithDialog( SMTPit_Append )
```

Result:

Appends the file you choose from an open file dialog to the list attachments for the current email.

SMTPit_SetAuthentication

Description

If your mail server requires authentication (requires you to log in) before it allows you to send email, then use the SMTPit_SetAuthentication function to assign the type of authentication your mail server requires. The types of authentication SMTPit supports are (in increasing levels of security): None (off), Plain, Login, and CRAM-MD5.

Return Type

Text

Format

SMTPit_SetAuthentication (**Type** ; **Username** ; **Password** ; **TLSType**)

Required Parameters

Type
The Type of Authentication (Specify None; Default; Plain; Login; CRAM-MD5; or Auto). Plain, Login, and CRAM-MD5 are the actual types of authentication the mail server supports. If you know which one your mail server supports, select it, otherwise use the Auto type. Auto will make the plug-in figure out which type works with your mail server. Specifying None will turn authentication off. Default uses the values set in the configuration dialog.

Optional Parameters

Username
Your account Username.

Password
Your account Password.

TLSType
The type of SSL/TLS/STARTTLS to use. (None; AfterConnect; or BeforeConnect) The type depends on your mail server. If you do not know the type your mail server requires, try both to see which type works.

Examples

Example 1

Code:

```
SMTPit_SetAuthentication( "none" )
```

Result:

Sets the authentication to "none"

Example 2

Code:

```
SMTPit_SetAuthentication( "PLAIN" ; "username" ; "password" )
```

Result:

Sets the authentication to "PLAIN"

Example 3

Code:

```
SMTPit_SetAuthentication( "LOGIN" ; "username" ; "password" )
```

Result:

Sets the authentication to "LOGIN"

Example 4

Code:

```
SMTPit_SetAuthentication( "CRAM-MD5" ; "username" ; "password" )
```

Result:

Sets the authentication to "CRAM-MD5"

SMTPit_SetBCC

Description

Use the SMTPit_SetBCC function to assign the "BCC" (Blind Carbon Copy) email address(es). Any email addresses you have in the BCC list will have the email sent to them, but their email will not show up in the actual email. If you have multiple email addresses and you want to set them all at once, separate each email address with a paragraph mark or, if in a field, with a return. SMTPit will validate the email addresses you give it to see if they "look" like email addresses. If SMTPit cannot find a valid email address, it will return an error. For examples of what SMTPit considers to be a "valid" email address, see the SMTPit_SetTo function.

Return Type

Text

Format

SMTPit_SetBCC (**BCC(s)** ; **SMTPit_Append**)

Required Parameters

BCC(s)
The BCC Email Addresses for the email.

Optional Parameters

SMTPit_Append
Leave this in to add the(se) Email Address(es) to the current list of Addresses.

Related Items

SMTPit_SetCC, SMTPit_SetFrom, SMTPit_SetReplyTo, SMTPit_SetTo

Examples

Example 1

Code:

```
SMTPit_SetBCC( "\"Doe, John\" <john@somewhere.com>" )
```

Result:

Adds the email "Doe, John" <john@somewhere.com> to the BCC field.

Example 2

Code:

```
SMTPit_SetBCC( "smtpit@nowhere.com¶info@nowhere.com" )
```

Result:

Adds the email addresses "smtpit@nowhere.com" and "info@nowhere.com" to the BCC field.

Example 3

Code:

```
SMTPit_SetBCC( "smtpit@nowhere.com" ; SMTPit_Append )
```

Result:

Appends the email address "smtpit@nowhere.com" to the current list of BCCs.

SMTPit_SetBody

Description

Use the SMTPit_SetBody function to set the Plain Text or HTML Body of the email. The "Body" of the email is the content of the email that the receiver will read.

Return Type

Text

Format

SMTPit_SetBody (**Content** ; **Type** ; **SMTPit_Append**)

Required Parameters

Content
The Plain Text or HTML content for the email.

Optional Parameters

Type
The Content Type; specify "Text" or "HTML". (Default is "Text".)

SMTPit_Append
Leave this in to add this Text to the end of the current Text content.

Related Items

SMTPit_SetBodyFooter, SMTPit_SetBodyHeader, SMTPit_SetBodyWithFile, SMTPit_StyledTextToHTML

Examples

Example 1

Code:

```
SMTPit_SetBody( MyEmailBodyField )
```

Result:

Uses the text in the "MyEmailBodyField" field for the text body of your email message.

Example 2

Code:

```
SMTPit_SetBody( "Hello World!" )
```

Result:

Uses the text "Hello World!" for the text body of your email message.

Example 3

Code:

```
SMTPit_SetBody( "Hello World!" ; "Text" ; SMTPit_Append )
```

Result:

Appends "Hello World!" to the current text body block.

Example 4

Code:

```
SMTPit_SetBody( MyHTMLEmailBodyField ; "HTML" )
```

Result:

Sets the HTML body to the html in the "MyHTMLEmailBodyField"

Example 5

Code:

```
SMTPit_SetBody( "<h1>Hello World!</h1>" ; "HTML" ; SMTPit_Append )
```

Result:

Appends the html "<h1>Hello World!</h1>" to the current HTML body.

SMTPit_SetBodyFooter

Description

This unique function will allow you to have a "Footer" section in the Plain Text or HTML Body of your email. Possible uses include an email signature, a public key, an email advertisement, quotes, or a passage telling how to unsubscribe from your list. You can also set a Default Footer in the Configuration Dialog. If you assign a Plain Text or HTML Footer using SMTPit_SetBodyFooter, SMTPit Pro will ignore the default footer.

Return Type

Text

Format

SMTPit_SetBodyFooter (**Content** ; **Type** ; **SMTPit_Append**)

Required Parameters

Content
The Plain Text or HTML content for the email Footer.

Optional Parameters

Type
The Content Type; specify "Text" or "HTML". (Default is "Text".)

SMTPit_Append
Leave this in to add Text or HTML to the end of the current Footer.

Related Items

SMTPit_SetBody, SMTPit_SetBodyHeader, SMTPit_SetBodyWithFile

Examples

Example 1

Code:

```
SMTPit_SetBodyFooter( FooterGlobal )
```

Result:

Uses the text in the "FooterGlobal" field as the footer to your email message.

Example 2

Code:

```
SMTPit_SetBodyFooter( "This is my footer" )
```

Result:

Uses the text "This is my footer" as the footer to your email message.

Example 3

Code:

```
SMTPit_SetBodyFooter( "Additional Footer" ; "Text" ; SMTPit_Append )
```

Result:

Appends the text "Additional Footer" to the current footer block.

Example 4

Code:

```
SMTPit_SetBodyFooter( "<p><hr><br>John Smith<br>john@somewhere.com</p>" ; "HTML" )
```

Result:

Sets the HTML footer to the HTML code above.

Example 5

Code:

```
SMTPit_SetBodyFooter( "<p><hr><br>John Smith<br>john@somewhere.com</p>" ; "HTML": SMTPit_A
```

Result:

Appends the HTML code above to the current HTML footer.

SMTPit_SetBodyHeader

Description

This unique function will allow you to have a "Header" section in the Plain Text or HTML Body of your email message. Possible uses are email letterheads or personalized email messages for mass emailing. SMTPit Pro also lets you to set a Default Header in the Configuration Dialog. If you assign a Plain Text or HTML Header using SMTPit_SetBodyHeader, SMTPit Pro will ignore the Default Header.

Return Type

Text

Format

SMTPit_SetBodyHeader (**Content** ; **Type** ; **SMTPit_Append**)

Required Parameters

Content
The Plain Text content for the email Header.

Optional Parameters

Type
The Content Type; specify "Text" or "HTML". (Default is "Text".)

SMTPit_Append
Leave this in to add this Text or HTML to the end of the current Header.

Related Items

SMTPit_SetBody, SMTPit_SetBodyFooter, SMTPit_SetBodyWithFile

Examples

Example 1

Code:

```
SMTPit_SetBodyHeader(  "Dear " & Contact Name & "¶¶" )
```

Result:

Sets the text header to "Dear", the name in the "Contact Name" field, and two returns.

Example 2

Code:

```
SMTPit_SetBodyHeader( "Additional Header Text" ; "Text" ; SMTPit_Append )
```

Result:

Appends the text "Additional Header Text" to the Text Header.

Example 3

Code:

```
SMTPit_SetBodyHeader( "<p>Dear " & Contact Name & ",</p>" ; "HTML" )
```

Result:

Sets the HTML header to "Dear" and the the name in the "Contact Name" field.

Example 4

Code:

```
SMTPit_SetBodyHeader( "<p>Dear " & Contact Name & ",</p>" ; "HTML" ; SMTPit_Append )
```

Result:

Appends "Dear" and the the name in the "Contact Name" field to the html header.

SMTPit_SetBodyWithFile

Description

You can use the SMTPit_SetBodyWithFile function to load a text file into the Plain Text or HTML Body of the email.

Return Type

Text

Format

SMTPit_SetBodyWithFile (**File** ; **Type** ; **SMTPit_Append**)

Required Parameters

File
The Path and File Name of the File to set as the Text or HTML content for the email.

Optional Parameters

Type
The Content Type; specify "Text" or "HTML". (Default is "Text".)

SMTPit_Append
Leave this in to add this Text or HTML to the end of the current content.

Related Items

SMTPit_SetBody, SMTPit_SetBodyFooter, SMTPit_SetBodyHeader, SMTPit_StyledTextToHTML

Examples

Example 1

Code:

```
SMTPit_SetBodyWithFile( "c:\My Documents\email body.txt" )
```

Result:

Uses the file "email body.txt" for the text body of your email message.

Example 2

Code:

```
SMTPit_SetBodyWithFile( "/Documents/email body.txt" )
```

Result:

Uses the file "email body.txt" for the text body of your email message.

Example 3

Code:

```
SMTPit_SetBodyWithFile( "/Documents/email body2.txt" ; "Text" ; SMTPit_Append )
```

Result:

Appends the file "email body2.txt" to the text body of your email message.

Example 4

Code:

```
SMTPit_SetBodyWithFile( "c:\My Documents\email body.htm" ; "HTML" )
```

Result:

Sets the HTML body to the contents of the "email body.htm" file on Windows.

Example 5

Code:

```
SMTPit_SetBodyWithFile( "/Documents/email body.htm" ; "HTML" )
```

Result:

Sets the HTML body to the contents of the "email body.htm" file on Mac.

Example 6

Code:

```
SMTPit_SetBodyWithFile( "/Documents/email body2.htm" ; "HTML" ; SMTPit_Append )
```

Result:

Appends the contents of the "email body2.htm" file to the HTML block to on Mac.

Example 7

Code:

```
SMTPit_SetBodyWithFile( "c:\My Documents\email body2.htm" ; "HTML" ; SMTPit_Append )
```

Result:

Appends the contents of the "email body2.htm" file to the HTML block to on Windows.

SMTPit_SetCC

Description

Use the SMTPit_SetCC function to assign the "CC" (Carbon Copy) email address(es). If you have multiple email addresses and you want to set them all at once, separate each email address with a paragraph mark or, if in a field, with a return. SMTPit Pro will validate the email addresses you give it to see if they "look" like email addresses. If SMTPit Pro cannot find a valid email address, it will return an error. For examples of what SMTPit Pro considers to be a "valid" email address, see the SMTPit_SetTo function.

Return Type

Text

Format

SMTPit_SetCC (**CC(s)** ; **SMTPit_Append**)

Required Parameters

CC(s)
The CC Email Addresses for the email.

Optional Parameters

SMTPit_Append
Leave this in to add the(se) Email Address(es) to the current list of Addresses.

Related Items

SMTPit_SetBCC, SMTPit_SetFrom, SMTPit_SetReplyTo, SMTPit_SetTo

Examples

Example 1

Code:

```
SMTPit_SetCC( "\"Doe, John\" <john@somewhere.com>" )
```

Result:

Adds the email "Doe, John" <john@somewhere.com> to the CC field.

Example 2

Code:

```
SMTPit_SetCC( "smtpit@nowhere.com¶info@nowhere.com" )
```

Result:

Adds the email addresses "smtpit@nowhere.com" and "info@nowhere.com" to the CC field.

Example 3

Code:

```
SMTPit_SetCC( "smtpit@nowhere.com" ; SMTPit_Append )
```

Result:

Appends the email address "smtpit@nowhere.com" to the current list of CCs.

SMTPit_SetCharSet

Description

Sets the Character Set that SMTPit uses when generating the email.

Return Type

Text

Format

SMTPit_SetCharSet (**CharSet**)

Required Parameters

CharSet
The Character Set for the email. See the SMTPit Configuration Dialog for valid values.

Example

Code:

```
SMTPit_SetCharSet( "UTF-8" )
```

Result:

Sets the email character set to UTF-8.

SMTPit_SetFrom

Description

Use the SMTPit_SetFrom function to assign the "From" email address. If you assign a from email address using this function, SMTPit Pro will ignore the Default From field in the Configuration Dialog. SMTPit Pro will validate the email address you give it to see if it "looks" like an email address. If SMTPit Pro cannot find a valid email address, it will return an error. For examples of what SMTPit Pro considers to be a "valid" email address, see the SMTPit_SetTo function.

Return Type

Text

Format

SMTPit_SetFrom (**From**)

Required Parameters

From
The From Email Address.

Related Items

SMTPit_SetBCC, SMTPit_SetCC, SMTPit_SetReplyTo, SMTPit_SetTo

Examples

Example 1

Code:

```
SMTPit_SetFrom( "smtpit@nowhere.com" )
```

Result:

Sets the from address to "smtpit@nowhere.com".

Example 2

Code:

```
SMTPit_SetFrom( "SMTPit Support <smtpit@nowhere.com>" )
```

Result:

Sets the from address to "SMTPit Support <smtpit@nowhere.com>".

Example 3

Code:

```
SMTPit_SetFrom( Company Name & "Support <" & Company Email & ">" )
```

Result:

Sets the from address to the text in the "Company Name" field and the email address in the "Company Email" field.

Example 4

Code:

```
SMTPit_SetFrom( "" )
```

Result:

Resets the from field so that it will use the default from address.

SMTPit_SetHost

Description

The Host is the domain name or IP address of the SMTP Pro mail server you are connecting to. You can assign the host in your scripts, or set up a Default Host in the Configuration Dialog. SMTPit_Send and SMTPit_Connect will return an error if no host has been set. If you assign a host with this function, SMTPit Pro will ignore the Default Host in the Configuration Dialog.

Return Type

Text

Format

SMTPit_SetHost (**Host** ; **Port**)

Required Parameters

Host
The Domain Name or IP Address of the SMTP Server to send the email through.

Optional Parameters

Port
The TCP/IP Port number for the SMTP Server. (Default is 25.)

Examples

Example 1

Code:

```
SMTPit_SetHost( "mail.yourdomain.com" )
```

Result:

Sets the email host to "mail.yourdomain.com".

Example 2

Code:

```
SMTPit_SetHost( "mail.yourdomain.com" ; "465")
```

Result:

Sets the email host to "mail.yourdomain.com", and sets the port to 465.

SMTPit_SetInlineImage

Description

The SMTPit_SetInlineImage function is similar to the SMTPit_SetAttachment function. This function allows you to attach a graphic image file from your hard drive or network to the current email being sent. SMTPit Pro takes this graphic and makes it an inline graphic image for you to use in your HTML email. You must specify the full path and filename of the image. You can assign more than one file with this function by separating the file names with the paragraph mark or, if in a field, with a return. You can also reference a container field to assign an image in it. If you have Inline Images assigned in the Configuration Dialog, they will always be sent with your email, and they will be added to the email first. If you do not assign an HTML Body, then an HTML email will not be sent and these Inline Images will be ignored.

Note, the Inline Images set with this function can only be used in the following HTML tags:

<img src="">
<input src="">
<body background="">
<table background="">
<tr background="">
<td background="">
<th background="">

Return Type

Text

Format

SMTPit_SetInlineImage (**Image(s)** ; **Type** ; **SMTPit_Append**)

Required Parameters

Image(s)
Specify the Path(s) and File Name(s) of the Inline Images(s) on the Hard Drive (separate multiple Inline Images with a Paragraph Mark or Return); or specify a Container Field that holds the Inline Image.

Optional Parameters

Type
The Inline Image Type; currently only "HTML" is supported. (Default is "HTML".)

SMTPit_Append
Leave this in to add the(se) Inline Images(s) to the current list of Inline Images.

Related Items

SMTPit_SetInlineImageWithDialog

Examples

Example 1

Code:

```
SMTPit_SetInlineImage( "c:\Images\logo.gif" )
```

Result:

Sets the "logo.gif" image as an Inline Image on Windows.

Example 2

Code:

```
SMTPit_SetInlineImage( "/Images/logo.gif" )
```

Result:

Sets the "logo.gif" image as an Inline Image on Mac.

Example 3

Code:

```
SMTPit_SetInlineImage( "/Thumbnails/product1.jpg¶/Thumbnails/product2.jpg" )
```

Result:

Sets the "product1.jpg" and "product2.jpg" images as Inline Images on Mac.

Example 4

Code:

```
SMTPit_SetInlineImage( "c:\Images\logo.gif" ; "HTML"; SMTPit_Append )
```

Result:

Appends the "logo.gif" image as an HTML Image on Windows.

Example 5

Code:

```
SMTPit_SetInlineImage( ImageContainerField )
```

Result:

Uses the image in the container field "ImageContainerField".

Example 6

Code:

```
SMTPit_SetInlineImage( ImageContainerField ; "HTML"; SMTPit_Append )
```

Result:

Appends the image in the container field "ImageContainerField".

SMTPit_SetInlineImageWithDialog

Description

The SMTPit_SetInlineImageWithDialog function is similar to the SMTPit_SetAttachmentWithDialog function. This function presents you with a standard open file dialog that lets you choose an image file from your hard drive. It then uses the chosen image as an inline image for you to use in your HTML emails.

Return Type

Text

Format

SMTPit_SetInlineImageWithDialog (**Type** ; **StartPath** ; **Prompt** ; **SMTPit_Append**)

Optional Parameters

Type
The Inline Image Type; currently only "HTML" is supported. Default is "HTML".

StartPath
The Path to the Folder to display when the dialog first opens.

Prompt
An optional Prompt or Message to display on the dialog.

SMTPit_Append
Leave this in to add this Inline Image to the current list of Inline Images.

Related Items

SMTPit_SetInlineImage

Examples

Example 1

Code:

```
SMTPit_SetInlineImageWithDialog
```

Result:

Opens a file dialog and uses the chosen file as an HTML Image.

Example 2

Code:

```
SMTPit_SetInlineImageWithDialog( ""; ""; ""; SMTPit_Append )
```

Result:

Opens a file dialog and appends the chosen file as an HTML Image.

SMTPit_SetPriority

Description

This function allows you to set the "Priority" of an email. Most email clients understand the Priority setting on an email and will highlight the email as more or less important than other emails. This function will take the numbers 1 through 5 with 1 being the Highest Priority, and 5 being the Lowest. This function will also take one of the following five words: "Highest", "High", "Normal", "Low", or "Lowest". The Priority defaults to 3 or "Normal".

Return Type

Text

Format

SMTPit_SetPriority (**Priority**)

Required Parameters

Priority
The Priority of the email. Use "Highest", "High", "Normal", "Low", "Lowest", or the numbers 1 through 5.

Examples

Example 1

Code:

```
SMTPit_SetPriority( 1 )
```

Result:

Sets the priority to "Highest".

Example 2

Code:

```
SMTPit_SetPriority( "High" )
```

Result:

Sets the priority to "High"

Example 3

Code:

```
SMTPit_SetPriority( Email Priority )
```

Result:

Sets the priority to the value in the "Email Priority" field.

SMTPit_SetReplyTo

Description

This function is used to set the "Reply-To" header of an email message. Most email clients will look for this email header and when a user replies to an email, the email will be sent to this email address.

Return Type

Text

Format

SMTPit_SetReplyTo (**ReplyTo**)

Required Parameters

ReplyTo
The Reply-To Email Address

Related Items

SMTPit_SetBCC, SMTPit_SetCC, SMTPit_SetFrom, SMTPit_SetTo

Example

Code:

```
SMTPit_SetReplyTo( "replyaddress@my.com" )
```

Result:

Sets the Reply-To to "replyaddress@my.com".

SMTPit_SetSubject

Description

Use this function to set the Subject of the email. If you assign a Subject of the email with this function, SMTPit Pro will ignore the Default Subject field in the Configuration Dialog.

Return Type

Text

Format

SMTPit_SetSubject (**Subject** ; **SMTPit_Append**)

Required Parameters

Subject
The Subject for the email.

Optional Parameters

SMTPit_Append
Leave this in to add this onto the end of the current Subject.

Examples

Example 1

Code:

```
SMTPit_SetSubject( "In response to your inquiry" )
```

Result:

Sets the subject to "In response to your inquiry".

Example 2

Code:

```
SMTPit_SetSubject( "In response to your inquiry" ; SMTPit_Append )
```

Result:

Appends the text "In response to your inquiry" to the current subject.

SMTPit_SetTo

Description

Use the SMTPit_SetTo function to assign the "To" email address(es). If you have multiple email addresses and you want to set them all at once, separate each email address with a paragraph mark or, if in a field, with a return. SMTPit Pro will validate the email addresses you give it to see if they "look" like email addresses. If SMTPit Pro cannot find a valid email address, it will return an error.

SMTPit Pro considers all of the following to be "valid" email addresses:

smtpit@nowhere.com
<smtpit@nowhere.com>
SMTPit Support <smtpit@nowhere.com>
"SMTPit Support" <smptit@nowhere.com>
"Smith, John" <john@nowhere.com>
<smtpit@nowhere.com> (SMTPit Support)

The basic rule is, if you have any "human readable" name in the email address (like "SMTPit Support"), then you need angled brackets (< >) around the actual email address. This is so mail servers and mail clients can correctly identify the email address. If you have a comma in the "human readable" name, then you need to enclose it in double quotes ("") (like "Doe, John").

Return Type

Text

Format

SMTPit_SetTo (**To(s)** ; **SMTPit_Append**)

Required Parameters

To(s)
The To Email Addresses for the email.

Optional Parameters

SMTPit_Append
Leave this in to add the(se) Email Address(es) to the current list of Addresses.

Related Items

SMTPit_SetBCC, SMTPit_SetCC, SMTPit_SetFrom, SMTPit_SetReplyTo

Examples

Example 1

Code:

```
SMTPit_SetTo( "SMTPit Support <smtpit@nowhere.com>" )
```

Result:

Sets the To email to "SMTPit Support <smtpit@nowhere.com>".

Example 2

Code:

```
SMTPit_SetTo( "smtpit@nowhere.com¶pop3it@nowhere.com")
```

Result:

Sets the To addresses to "smtpit@nowhere.com" and "pop3it@nowhere.com".

Example 3

Code:

```
SMTPit_SetTo( "sales@nowhere.com" ; SMTPit_Append )
```

Result:

Appends the address "sales@nowhere.com" to any To addresses that have already been set.

SMTPit_StatusWindow

Description

Use this function to show a status window that displays information about what the plug-in is currently doing. When you are sending an email, the status window will show you what part of the email it is currently sending as well as a progress bar indicating how much of that part it has completed. Default placement of the status window is the center of the screen.

If you want to specify a starting location for the status window, specify the coordinates of the left and top of the dialog in pixels. If you wanted to display it in the top right hand corner of your screen, and your screen resolution is set to 800x600, you could specify left as "700" and top as "100". If you specify a negative one ("-1") as either the x (across) or y (down) coordinates, the status window will be centered on that axis. For instance, if you want to display it on the bottom of your screen in the center, you would specify left as "-1" and top as "600"; or in the center of the screen by specifying left as "-1" and top as "-1".

Return Type

Text

Format

SMTPit_StatusWindow (**Action** ; **Left** ; **Top** ; **WindowTitle**)

Required Parameters

Action
The action for the Status Window. Specify one of the following:

Show - This will display the Status Window.
Hide - This will hide the Status Window.
State - This tells you if the Status Window is currently shown or hidden. This will return the state as well as the current Left and Top coordinates of the Status Window.
Move - Use this for the Action parameter and the Left and Top parameters to move the Status Window to the coordinates you define.

Optional Parameters

Left
The horizontal screen coordinate for the Status Window. (Use -1 to Center on this axis.)

Top
The vertical screen coordinate for the Status Window. (Use -1 to Center on this axis.)

WindowTitle
A Custom Title for the Status Window. (Only valid with the "Show" action.)

Examples

Example 1

Code:

```
SMTPit_StatusWindow( "Show" )
```

Result:

Displays the status window in the center of the screen.

Example 2

Code:

```
SMTPit_StatusWindow( "Show" ; 700 ; 100 )
```

Result:

Opens the status window 700 pixels from the left, and 100 pixels from the top.

Example 3

Code:

```
SMTPit_StatusWindow( "Show" ;  -1 ; 600 )
```

Result:

Opens the status window 600 pixels from the top, and centers it from left to right on the screen.

Example 4

Code:

```
SMTPit_StatusWindow( "Move" ;  100 ; 100 )
```

Result:

Moves the status window 100 pixels from the top, and 100 pixels from left of the screen.

Example 5

Code:

```
SMTPit_StatusWindow( "Hide" )
```

Result:

Hides the status window.

Example 6

Code:

```
SMTPit_StatusWindow( "State" )
```

Result:

Returns the state the Status Window as either shown or hidden, and the current location of the

Status Window.

SMTPit_StyledTextToHTML

Description

This function will convert styled text to HTML. In other words, you can convert a field that has styled text such as bold, underline, color, font, size, etc. to html and in effect, send HTML email without the end user knowing any HTML.

The styled text can be converted to CSS based HTML or into old style HTML tags. CSS is more precise, however not all email clients will render CSS. It will be up to you as the solution developer to decide which type will fit the needs of your recipients.

Return Type

Text

Format

SMTPit_StyledTextToHTML (**FieldName** ; **UseCSS** ; **EncodeAngledBrackets** ; **IgnoreTraits**)

Required Parameters

FieldName
The fully qualified Field Name of the Filed that contains the Styled Text. (Note: this must be in quotes, and be in the form "databasename::fieldname".)

Optional Parameters

UseCSS
Set to True to use CSS or False to use older <font> tags. (Default is False.)

EncodeAngledBrackets
Set to True to encode any angled brackets ('<' and '>') found in the field. If you have HTML already in the field, you probably do not want this option. (Default is False.)

IgnoreTraits
Define which traits to ignore. (Use the values 'Font', 'Face', 'Size', 'Color'. You can define more than one.)

Related Items

SMTPit_SetBody, SMTPit_SetBodyFooter, SMTPit_SetBodyHeader

Examples

Example 1

Code:

```
SMTPit_StyledTextToHTML( "databasename::fieldname" )
```

Result:

Will convert the styled text in the databasename::fieldname field using the old style tag method.

Example 2

Code:

```
SMTPit_StyledTextToHTML( "databasename::fieldname" ; True )
```

Result:

Will convert the styled text in the databasename::fieldname field using the CSS method.

Example 3

Code:

```
SMTPit_StyledTextToHTML( "databasename::fieldname" ; False )
```

Result:

Will convert the styled text in the databasename::fieldname field using the old style tag method.

Example 4

Code:

```
SMTPit_StyledTextToHTML( "databasename::fieldname" ; False ; True )
```

Result:

Will convert the styled text in the databasename::fieldname field using the old style tag method and encode angle brackets.

Example 5

Code:

```
SMTPit_StyledTextToHTML( "databasename::fieldname" ; False ; False ; "Size")
```

Result:

Will convert the styled text in the databasename::fieldname field using the old style tag method, and will ignore any font sizes.

Example 6

Code:

```
SMTPit_StyledTextToHTML( "databasename::fieldname" ; True ; False ; "Size¶Color")
```

Result:

Will convert the styled text in the databasename::fieldname field using the CSS method, and will ignore font size and color.

SMTPit_VerifyEmailAddress

Description

You can use this function to verify the syntax of an email address to see if it "looks" correct. You can also verify the email address against its host server to see if it is valid.

Note: The "Server Verification" feature of this function should be used with caution and at your own risk. There is only one method for doing this sort of verification, and it has unfortunately been abused by "spammers". For this reason, most mail servers have disabled this sort of verification, or if they have not disabled it, they have added some security to blacklist computers that constantly attempt to verify email addresses. So, to keep yourself from being blacklisted, it would be a good idea to only verify an email address once, or at the very most, only verify an email address every month or two. You should also keep in mind that if this function returns an error, it does not necessarily mean that the email address is invalid. It could simply mean that the mail server has disabled this sort of verification.

Return Type

Text

Format

SMTPit_VerifyEmailAddress (**CheckType** ; **EmailAddress** ; **Timeout**)

Required Parameters

CheckType
The type of Verification Check to do. (Specify "Syntax" to check to see if the EmailAddress "looks" like an email address; this is the Default. Specify "Server" to Verify the email address against its host server.)

EmailAddress
The Email Address to Verify.

Optional Parameters

Timeout
The number of seconds to wait before giving up when doing a Server Verification. (The 'Connection Timeout' setting in the Configuration Dialog is the Default Setting.)

Examples

Example 1

Code:

```
SMTPit_VerifyEmailAddress( "Syntax" ; "sales@mydomain.com" )
```

Result:

Verified.
sales@mydomain.com

Example 2

Code:

```
SMTPit_VerifyEmailAddress( "Syntax" ; "sales@mydomaincom" )
```

Result:

ERROR: VerifyEmailAddress: Email failed syntax check. (It doesn't "look" like an email address.)

Example 3

Code:

```
SMTPit_VerifyEmailAddress( "Server" ; "sales@mydomain.com" )
```

Result:

ERROR: VerifyEmailAddress: SendRcptToCmd: Mail Server Error: unrouteable address

SMTPit_Version

Description

This function returns the current version of SMTPit Pro. This function is useful for testing whether or not the plug-in is installed and enabled. If you call this function and a question mark ("?") is returned, then the plug-in is either not installed or not enabled.

Return Type

Text

Format

SMTPit_Version

Related Items

SMTPit_VersionAutoUpdate

Example

Code:

```
SMTPit_Version
```

Result:

Returns the SMTPit Pro version like "SMTPit Pro v.4.1.12".

SMTPit_VersionAutoUpdate

Description

This function returns an Auto Update friendly Version number of SMTPit Pro. The format of this version number is always exactly 8 digits long. The first two digits represent the major version of the plug-in (zero-filled). The third and fourth digits represent the minor version of the plug-in (zero-filled). The fifth and sixth digits represent the update portion of the version (zero-filled). The final two digits represent a special build number or a beta version number and will usually be zeros.

As an example, for SMTPit 4.1.12, the major version is 4, the minor version is 1, the update number is 12, and there is no special build or beta version defined. So, the resulting Auto Update friendly version number would be 04011200.

Return Type

Number

Format

SMTPit_VersionAutoUpdate

Related Items

SMTPit_Version

Example

Code:

```
SMTPit_VersionAutoUpdate
```

Result:

Returns 04011200 for SMTPit Pro version 4.1.12.