BadMailManager.com

BMM64_DLL.dll

DLL User's Guide

Published by
Nectare Ltd
Contact: admin@nectare.co.uk

Author
Peter JA Noblett

Last Revision
20th February 2017

The Legal Bits

GOOD NEWS – THE SAMPLE BADMAILMANAGER SOURCE CODE AVAILABLE ON THE BADMAILMANAGER.COM WEBSITE IS SUBJECT TO THE FOLLOWING STANDARD MIT SOFTWARE LICENSE

NB. The "associated documentation files" refers to the notes relating to the sample code which are also available on the BadMailManager.com website.

This document is
Copyright © 2017 Nectare Ltd. All rights reserved.
And therefore subject to standard copyright terms.

Product Registration & Licensing

Do I need buy a license?

No - If you use the BMM64.exe in conjunction with the BMM64_DLL.dll in the form that can be downloaded from our website then **you do not need purchase a license**. However you must agree to the terms of the license agreement that is displayed when the product is installed.

Yes - To use the BMM64_DLL.dll within your own project you will need to purchase a license.

Why? - We are a commercial organisation and BMM64.exe has been provided to enable you to evaluate our BMM64_DLL.dll. If you like what it does you may wish to incorporate the dll into your own support, data cleaning and reporting functions. This would be cheaper and quicker writing, testing and maintaining your own code.

Please contact *sales@nectare.co.uk for details on licensing.*

Introduction

This guide lists and explain each of the functions built into the dll such that the dll can be incorporated into your own application. The source code of sample VB.NET and ASP.NET applications are available on the BadMailManager.com website.

The BadMailManager dll (BMM64.dll) is written in VB.NET but the primary functions have been written so that they can be easily called from any other .NET application, whether written in VB.NET, C#.NET or ASP.NET. See individual functions for details of output format.

Please read **the first 5 pages** of this guide **before** starting to use any of the sample code on our website http://badmailmanager.com – RTFM to avoid becoming a PICNIC!

Parameters

To generate output you will need to specify some common parameters plus some additional parameters specific to the type of report / output required.

Dlls are loaded into the same memory space as the exe that calls them. This enables you to load the common parameters as a collection prior to calling the required function. This simplifies software development as you have one, common block, for the parameters and a much shorter set of code to call a specific function. See Appendix A for a list of the standard parameters and Appendix B for the parameters for advanced searches.

File Cleaning / Delete Matching

As well as generating a .BAD file IIS may create associated files relating to the failed message, these can include:

- BDP files - Microsoft Exchange Diagnostic Message – This is a short non text file that will include a pointer to the original eml file (e.g. C:\inetpub\mailroot\Queue\NTFS_8a25d7bc01d1e8730000aa94.EML)
- BDR files – text files containing the body of the Non Delivery Report (NDR)

When archiving and / or deleting files there is an option to delete matching files (i.e. BDP and BDR files). If the delete matching option is selected any file with the same basename (i.e. file name less the extension) will be deleted as long as it's extension is not in the list of extensions being searched for.

E.g.

If archiving files with a .bad or .bdr extensions then

the parameter **BMM_fileext** would be ***.bad|*.bdr**

If delete matching=true, only files with the same base name but whose extension is not in **BMM_fileext** will be deleted. I.e. the .BDP files would be deleted.

HTML Output

Where a function returns an HTML data stream and there are matching files, the data will be embedded in a table. If there is an exception the HTML will be returned as a caption. (See Appendix H for HTML Exception Messages).

In all cases there are no <html></html> tags included to make it easy to insert the HTML in your pages.

Logging

Some functions have a logging option, if enabled the names of files:

- Archived
  If delete matching enabled also names of any matching files that were deleted

- Copied

- Deleted
  If delete matching enabled also names of any matching files that were deleted

Details are written to the log file **BMM64_log.csv** in the folder specified in the parameter **BMM_log**.

The log file is a text file in CSV format, hence the .csv extension.

Logging of CSV files

If a report is created as a CSV file (see BMM64_ParamsFileGetAndRun and BMM64_ReportAsCSVFile) the name of the CSV file created and the number of records in the report can be written to the log file.

To Enable Logging

Set the **Log** parameter in the function call to **-1** (true).

To Disable Logging

To disable logging set Log = 0.

NB. If you enable logging then ensure users has sufficient rights to create and write to the log file.

BMM_FileId – Encrypted File Names

For security the plain file names are not used when archiving, copying, deleting or viewing files. It might be considered dangerous to have an "open" dll sitting on a server which could be used for deleting / moving files. For security these routines are expecting the path + file name to be passed in a specific encrypted format, which is referred to to as the BMM_FileID. These routines will not work if passed the file name in plain text.

Run OK and Error Handling

Errors can happen, so we provide a simple and quick way to check whether a routine has worked OK or if there has been a problem.

When a function returns a string value the first two characters will be normally be "OK" (see individual function descriptions for any exceptions).

Where the function returns a collection of collections the last collection stores the status data.

The first element of the last collection will be "OK" if one or more matching files were found and all those found were successfully processed. "NOMATCH" if no matching files.

If there are no files with the specified extension(s) then "NOFILES" will be returned.

For those functions that require the common parameters and they are incorrect or incomplete the status returned will be "NOTVALID". The second element will be a collection of collections (one collection for each parameter). You will need to interrogate that collection to find the invalid item. (See Appendix D).

Your validation code would look something like:

```
If YourResult(YourResult.count)(1) <> "OK" Then
      If YourResult(YourResult.count)(1) <> "NOMATCH" then
            If YourResult(YourResult.count)(1) <> "NOFILES" then
                  If YourResult(YourResult.count)(1) = "NOTVALID" then
                        THE PARAMETERS YOU HAVE SET ARE INCORRECT
                  Else
                        'OOPS! Something has gone wrong
                  YOUR ERROR HANDLER
                  End If
            End If
      End If
End If
```

If there is an error the first element (1) of the status collection will be:

"ERR" - This is a managed error, i.e. unexpected, binary data, in a file
"ERROR" - An unexpected error, frequently due to a problem with a user's rights to a particular file of folder.

Pre-Requisites

Where files are archived, copied, created or deleted you need to ensure the user has sufficient rights otherwise you will get an error 75 type of message.

List of Main Functions (in Alphabetical Order)

BMM_ArcByFileId * $
Archives a file by moving it an archive folder

BMM_ArcCopyDelMatching #
Archive, Copy, Delete matching files

BMM_CopyByFileId * $
Copies a file from the source folder to another folder

BMM_DeleteByFileId * $
Deletes a file and if required, all associated files

BMM_EmptyCollA
Used to generate empty collection to hold basic and advanced parameters

BMM_Engine #
Generates details of matching files as collection of collections

BMM_LogReport $
Reads log file and generates a report as collection

BMM64_ParamsFileGet
Reads and returns a BMM parameters file as collection

BMM64_ParamsFileGetandRun $
Reads a BMM parameters file and runs the BMM_Engine

BMM64_ParamsFileGetAsDictionary $
Returns a BMM parameters file as Collections.Generic.Dictionary

BMM64_ParamsFileWrite
Generates a BMM parameters file

BMM_ReportAsCSVFile *
Calls BMM_Engine and creates a CSV file

BMM_ReportAsHTML *
Calls BMM_Engine and creates a report in HTML format

BMM_StatisticsAsHTML #
Generates HTML list of "from" addresses with the no of files per address

BMM_Validate #
Validates the primary parameters

BMM_ViewAFile *

Notes:
* These use the BMM_FileID
These require the report parameters to be built and passed to the dll prior to being called
$ All compatible with VB.NET, these should be compatible with C# as return string or a .net collection

List of Utility Functions (in Alphabetical Order)

BMM64_EnsureBackSlash
Used to ensure a folder names end in a backslash

BMM64_LogFileDelete
Deletes a log file

BMM64_ParamsNumber
Returns the number of standard parameters this version of the DLL is expecting (currently 18) this is used by BMM64.exe when building a BMM parameters file

BMM64_SECOP_SimpleEncrypt
A quick encryption routine used to generate and de-encrypt BMM_FileIDs

BMM64_Version
Returns the version number of the dll.

Appendices

BMM_ArcByFileId

Moves a specific file from the source folder (**BMM_source**) to an archive folder.

Function

```
Function BMM64_ArcByFileId(psBMM_FileID As String, psArchiveFolder As String,
piLog As Integer, psBMM_Log As String, pbDelMatching As Boolean, psBMM_fileext
As String) As String
```

Options

1) File Cleaning using Delete Matching
2) Logging

Pre-Requisites

None

Call

```
BMM64_ArchiveAFile (BMM_FileID, ArchiveFolder, Log, BMM_Log, DelMatching,
BMM_Fileext)
```

BMM_FileID

Full path and name of file to be archived in BMM64 encrypted format

ArchiveFolder

Folder where file is to be moved to (normally **BMM_archive**)

Log

See Logging

BMM_log

Folder where log file **BMM_Action.log** is written (normally that held in the **BMM_log** parameter) . If **Log** = 0 this field is not referenced and should be left as blank ("").

DelMatching

See Delete Matching

BMM_FileExt

Value as **BMM_fileext**, required when DelMatching = True to ensure files with extensions to be included in this report are not deleted.

Output *(as string with pipe" |" as the delimiter)*

If successful:

"OK|FileName"

If failed due to bad BMM_FileID:

"BADID|" + BMM_FileID passed to the function

If something has gone wrong:

"ERROR|" + description of the error

BMM_ArcCopyDelMatching

Can be used for:

- Archiving
- Copying
- Deleting

matching files.

Function

```
Function BMM64_ArcCopyDelMatching(piAction As Integer, piLog As Integer) As
Collection
```

Options

1) Action
2) Logging

Pre-Requisites

Requires basic BMMColA parameters and, when required, the advanced search parameters, BMMColB, to be created prior to being called
NB. Basic parameters at Appendix A, advanced parameters at Appendix B.

Call

```
BMM64_ArchiveMatching(Action, Log)
```

Action

31 Archive* – Moves each file to the designated archive folder
32 Copy
38 Delete*

* If the delete matching flag **BMM_delmatch** is set on in the parameters any associated files will also be deleted

Log

See Logging

Output *(as collection of collections Type: Microsoft.Visual.Basic.Collection)*

One collection for each successfully processed file. Each collection contains an element for each column selected to be included in the report (Appendix C).

The last collection contains the status.

If one or more matching files were found:

1. "OK"
1. "BMM64_ArcCopyDelMatching"
2. Time taken to process the records, in seconds
3. Number of files read
4. Number of matching files
5. Number of files processed
6. "Archive / Copy / Delete files"
7. "999 files archived /copied /deleted"

If no matching files were found:

1. "NOMATCH"
2. "BMM64_ArcCopyDelMatching"
3. Time taken to process the records, in seconds
4. Number of files read
5. 0
6. 0
7. "Archive / Copy / Delete files"
8. "No Matching Files"

If no files with the specified extension were found:

1. "NOFILES"
2. "BMM64_ArcCopyDelMatching"
3. Time taken to process the records, in seconds
4. 0
5. 0
6. 0
7. "Archive / Copy / Delete files"
8. "No files matching [BMM_fileext]"

If the parameters are invalid:

1. "NOTVALID"
2. "BMM64_ArcCopyDelMatching"
3. The collection, generated by **BMM_Validate**, that contains details of the parameters with any error messages (see **BMM_Validate**). If you need to interrogate this collection of collections, it is the same as element 2 from **BMM_validate**.

If a problem processing a specific file the routine will exit with the following:

1. "ERR"
2. "BMM64_ArcCopyDelMatching"
3. BMM64 error number (e.g. "M1011")
4. Number of files read
5. Number of matching files
6. Number of files processed
7. "Unable to Archive / Copy / Delete file x"
8. System generated error number & description
9. If error number = 75 message "You may not have sufficient rights..."
   else
   Some diagnostics information to assist our support team

If something has gone wrong:

1. "ERROR"
2. "BMM64_ArcCopyDelMatching" or "BMM64_Engine"
3. BMM64 error number (e.g. M1021, M1022)
4. System generated error number|error description
5. Diagnostics information to assist our support team

BMM_CopyByFileId

Copies a file from the source folder (**BMM_source**) to an archive folder.

Function

```
Function BMM64_CopyByFileId(psBMM_FileID As String, psArchiveFolder As String,
piLog As Integer, psBMM_Log As String) As String
```

Options

Logging

Pre-Requisites

None

Call

```
BMM64_CopyByFileId (BMM_FileID, ArchiveFolder, Log, BMM_Log)
```

BMM_FileID
Path and name of file to be copied in BMM64 encrypted format.

ArchiveFolder
Folder where file is to be copied to (normally **BMM_archive**)

Log
See Logging

BMM_log
Folder where log file **BMM_Action.log** is written (normally that held in the **BMM_log** parameter) . If **Log** = 0 this field is not referenced and should be left as blank ("").

Output *(as string with pipe" |" as the delimiter)*

If successful:
"OK|FileName"

If failed due to bad BMM_FileID:
"BADID|" + BMM_FileID passed to the function

If something has gone wrong:
"ERROR|" + description of the error

BMM_DeleteAFilebyId

Deletes a file from the source folder (**BMM_source**).

Function

```
Function BMM64_DeleteByFileId(psBMM_FileID As String, piLog As Integer,
psBMM_Log As String, pbDelMatching As Boolean, psBMM_fileext As String) As
String
```

Options

1) File Cleaning using Delete Matching
2) Logging

Pre-Requisites

None

Call:

```
BMM64_DeleteAFile (BMM_FileID, Log, BMM_Log, DelMatching, BMM_fileext)
```

BMM_FileID
Path and name of file to be deleted, passed in encrypted format.

Log
See Logging

BMM_log
Folder where log file **BMM_Action.log** is written (normally that held in the **BMM_log** parameter) . If **Log** = 0 this field is not referenced and should be left as blank ("").

DelMatching
See Delete Matching

BMM_FileExt
Value as **BMM_fileext**, required when DelMatching = True to ensure files with extensions to be included in this report are not deleted.

Output *(as string with pipe" |" as the delimiter)*

If successful:
"OK|FileName"

If failed due to bad BMM_FileID:
"BADID|" + BMM_FileID passed to the function

If something has gone wrong:
"ERROR|" + description of the error

BMM_EmptyCollA

Function

Creates an empty collection to be filled with the parameters required to use various functions.

Pre-Requisites

None.

Input

See Appendices A and B for list of fields.

Output (an empty Microsoft.Visual.Basic.Collection)

Use BMM_Validate to test the parameters in EmptyCollA are valid.

BMM_Engine

This contains the core code that interrogates the files. This routine returns a collection of collections, one collection for each matching file.

The parameters passed in via the collection BMMColA are always validated within this routine before the "engine" processing starts. There can be a range of conditions returned in the status (last) collection which needs to be checked whenever this function is called.

Function

```
Function BMM64_Engine() As Collection
```

Pre-Requisites

BMMColA needs to be created prior to being called.
NB. BMMColA holds the parameters required to run the BMM Engine

Call

```
BMM64_Engine()
```

Output (as collection of collections Type: Microsoft.Visual.Basic.Collection)

One collection for each matching file. Each collection contains an element for each column selected to be included in the report (Appendix C).

If one or more matching files were found:

1. "OK"
2. "BMM64_Engine"
3. Number of files read
4. Number of matching files
5. Time taken to run process, in seconds
6. If the link column (2) has been selected to be included in the report then this is the physical column that the link is in – needed to quickly add the links into reports, saves having to re-process the basic parameters.
7. If the file size column (71) has been selected to be included in report this physical column the file size is in – needed, as per 6 above.

If there no matching files found:

1. "NOMATCH"

2 - 7 as above

If no files with the specified extension were found:

1. "NOFILES"
2. "BMM64_Mod1_Engine"

If the parameters are invalid:

1. "NOTVALID"
2. "BMM64_Mod1_Engine"
3. The collection generated by **BMM_Validate**, this contains details of all the parameters that were validated together with any error messages (see **BMM_Validate**).

If something has gone wrong:

1. “ERROR”
2. BMM error number I.e.
   1. M1021 error in validation routine
   2. M1022 error in engine
3. System generated error number & error description
4. Diagnostics information to assist our support team

BMM_LogReport

Generates a list of log entries made as a System.Collections.Generic.List(Of String)

Function

```
Function BMM64_LogReport(psLogFile As String, peFrom As Date, peTo As Date,
psOptions As String) As Collection
```

Options

Logging

Pre-Requisites

None

Call

```
BMM64_LogReport(LogFile, DateFrom, DateTo, Options)
```

LogFile

Path + File Name of log file
If you pass **BMM_log**, the name of the folder where the log files are kept then the default log file name of "BMM64_log.csv" will be used;

DateFrom

The report to include all entries written on or after this date / time.
Format dd/mm/yyyy or dd/mm/yyyy HH:MM

DateTo

The report to include all entries up to and including this date / time.
Format as DateFrom

Options

Not used

Output *(as a collection*
Type: System.Collections.Generic.List(Of String))

A collection for every valid log as a string.
The last collection holding the status,

If successful:

1. "OK"

If something has gone wrong:

1. "ERROR|Error description"

BMM_ParamsFileGet

You can save the parameters to a special text file called a BMM parameters file. This routine reads BMM parameters file and returns the contents as a collection. You can then use / modify this data to run another report. See Appendix F.

Function

```
Function BMM64_ParamsFileGet(psFileName As String) As Collection
```

Options

None

Pre-Requisites

None

Call

```
BMM64_ParamsFileGet(FileName)
```

FileName

Path + file name of the BMM Parameters file.

Output *(as Microsoft.Visual.Basic.Collection)*

Example at Appendix F

BMM_ParamsFileGetAndRun

This loads a BMM parameters file and runs it.
NB. This routine ONLY works with the following actions:

0 =	Just validates the parameters
5 =	Returns the statistics (list of from addresses) inside an HTML table
15 =	Returns a report inside an HTML table
21 =	Generates report as a CSV file
41 =	Generates a list of BMMFileIds - each separated by vbCrlf. The first line reads "999 BMMFileIds for matching files"

Any other value for BMM_action parameter will return the message "Invalid action – must be 5, 15, 21 or 41 – See DLL User's Guide."

NB. This routine is used as the basis of BMM64Console.exe

Function

```
Function BMM64_ParamsFileGetAndRun(psFileName As String, pbActionOverRide As
Boolean, piNewAction As Integer, piLog As Integer) As String
```

Options

ActionOverRide – The current action setting can be overridden by setting this value to true and setting piNewAction to the new value.

Logging

Pre-Requisites

None

Call

```
BMM64_ParamsFileGetAndRun(FileName, ActionOverRide, NewAction, Log)
```

FileName
Path + file name of the BMM Parameters file.

ActionOverRide
True or false, if true need to pass a valid value in NewAction

NewAction
A valid action value of 0, 5, 15, 21, or 41 (Appendix A)

Log
See Logging

Output

Action

0 "OK" or description of error in parameters

5 Returns the statistics (list of from addresses) inside an HTML table.
(See HTML Output)

15 Returns a report inside an HTML table. (See HTML Output)

21 Generates report as a CSV file
The fields in each line will depend on which columns have been selected.
Plus additional last Line:
    "OK" – If matching files
Else
    "NOMATCH" – no matching files
    "NOFILES" – no files with specified extension(s)
    "NOTVALID" – invalid parameters
    If something has gone wrong "ERROR|" + error message

41 List of the BMMFileIds of matching files, each separated by vbCrlf.
Plus additional last Line:
    "OK" – If matching files
Else
    as 21 above

BMM_ParamsFileWrite

Writes a BMM parameters file using the data held in the parameters collection BMM_EmptyCollA. Sample file layout at Appendix F

Function

```
Function BMM64_ParamsFileWrite(psFileName As String, psNote As String) As
String
```

Options

None

Pre-Requisites

None

Call

```
BMM64_ParamsFileWrite(FileName, Notes)
```

FileName

Path + file name of the BMM Parameters file.

Notes

You can store notes with the parameters.

Output *(as string)*

If successful:

"OK"

If a problem

"ERROR|error description"

BMM_ReportAsCSVFile

Creates a report as a CSV file. The file is created with a hashed file name. See HTML in BMM_Reports.asp to see how to download it with a meaningful name.

Function

```
Function BMM64_ReportAsCSVFile(piLog As Integer) As String
```

Options

Logging

Pre-Requisites

BMMColA needs to be created prior to being called
NB. BMMColA holds the parameters required to run the BMM Engine

If the report is to be downloaded via the Internet the folder specified in BMM_CSV must be accessible from your web site.

Call

```
BMM_ReportAsCSVFile(Log)
```

Log

>0=Records in **BMM_Action.log**:
a) Path + name of CSV file
b) Number of records in the CSV file

Output

1. CSV file with headers and last line of "End Of Report - 999 matching files". The columns will be as set in the parameter **BMM_columns**. If using this CSV file as part of a back office cleaning process simply test for "End Of Report" in the last line to ensure CSV file created OK.

2. A collection:

   If successful:
   1. "OK"
   2. "X matching files"
   3. Name of CSV file

   If the parameters are invalid:
   1. "NOTVALID"
   1. "Invalid Parameters"

   If something has gone wrong:
   1. "ERROR"
   2. "Unable to generate CSV file"
   3. Diagnostics – *Procedure*
   4. Error number & description
   5. Diagnostics information to assist our support team

BMM_ReportAsHTML

Returns a HTML report as a table in plain text. This can then be embedded in a web page.

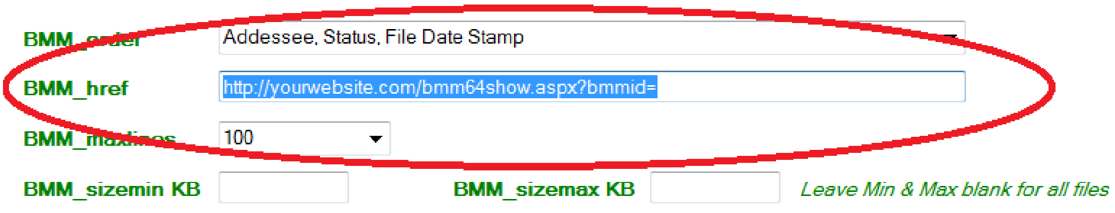
Function

```
Function BMM64_ReportAsHTML() As String
```

Pre-Requisites:

BMMColA needs to be created prior to being called.
NB. BMMColA holds the parameters required to run the BMM Engine

If the **Link To View File** column (Appendix C) is selected you will need to to enter the href that will be used when the link is activated.



BMM_order Addessee, Status, File Date Stamp

BMM_href http://yourwebsite.com/bmm64show.aspx?bmmid=

BMM_maxlines 100

BMM_sizemin KB

BMM_sizemax KB

Leave Min & Max blank for all files

The href declares the web page to be called.

```
e.g. http://yourwebsite/bmm64show.aspx?bmmid=
```

The first part "`bmm64show.aspx`" is the webpage to be called. The second, "`?bmmid=`" is the query string component used to pre-fix the BMM_FileID of the file to be viewed. If writing your own viewer it is recommended to stick with "`?bmmid=`".

The simplest way to ensure the BMM64 dll can be found is to put a copy of BMM64_DLL.dll in the /bin folder of your asp.net website.

Input:

BMM_ReportAsHTML()

Output *(text):*

If successful:

An HTML Table.

If the parameters are invalid:

"<caption style='font-size:150%'>Incorrect Parameter Settings</caption>"

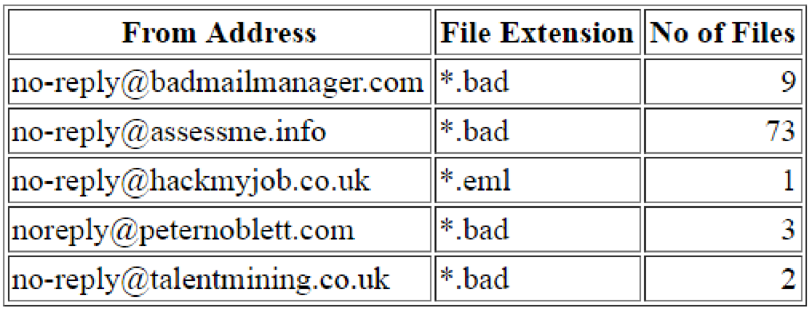
If something has gone wrong:

"<caption style='font-size:150%'>Unable to run report [Problem: Error *details of error*]</caption>"

NB. There are no <html></html> tags included to make it easier to insert the HTML data in your page.

BMM_StatisticsAsHTML

Reads matching files in the BMM_source folder and identifies each "from" address and counts the number of files for each "from" address + file extension pair. The results are returned as an HTML table.



From Address	File Extension	No of Files
no-reply@badmailmanager.com	*.bad	9
no-reply@assessme.info	*.bad	73
no-reply@hackmyjob.co.uk	*.eml	1
noreply@peternoblett.com	*.bad	3
no-reply@talentmining.co.uk	*.bad	2

Sample Statistics Report Function

```
Function BMM64_StatisiListFromAsHTML(psHeadings As String) As String
```

Option

Column headings are optional and delimited by | (pipe). The standard text is "From Address| File Extension|No of Files".

Pre-Requisites

BMMColA needs to be created prior to being called, ensure parameter BMM_action = 5
NB. BMMColA holds the parameters required to run the BMM Engine

Call

```
BMM64_ListFromAsHTML(ColHeadings)
```

ColHeadings

The three column headings delimited by | (pipe), see **Option** above.

Output *(as text)*

If successful:

An HTML Table

If the BMM_action <> 5

"<caption style='font-size:150%'>Incorrect Parameter Setting Action must be 5</caption>"

If the parameters are invalid

"<caption style='font-size:150%'>Incorrect Parameter Settings</caption>"

If something has gone wrong

"<caption style='font-size:150%'>ERROR M1041/2 - Unable to run report [Details: system generated *error number* | *error description*]</caption>"

BMM_Validate

Validates the parameters written to the collection BMM_EmptyCollA.

Function

```
Function BMM64_Validate() As Collection
```

Pre-Requisites

BMM_EmptyCollA has to be called to create a new collection within the dll memory space and then that collection need to be filled.

Call

```
BMM_Validate()
```

Output (collection of collections Type: Microsoft.Visual.Basic.Collection)

One collection for each parameter + final collection holds the status fields.

The collection for each parameter contains the following items:

1. Parameter i.e. "BMM_source"
2. The value you set
3. 0 (zero) if valid, else an error number
1. Error message describing why your parameter is invalid

NB. If NOTVALID elements 3 and or 4 of each sub collection need to be interrogated to identify the error.

The last collection:

If successful:

1. "OK" or "NOFILES"
2. No of files found that matching the specified extensions

If an invalid parameter found

1. "NOTVALID"
2. 0 (Zero)

BMM_ViewAFile

This returns the contents of a file as a block of data within a collection so that it can be displayed on a web page etc.

Function

```
Function BMM64_ViewAFile(psBMM_FileID As String) As Collection
```

Pre-Requisites

None

Call

```
BMM_ViewAFile (psBMM_FileID)
```

psBMM_FileID The full path of file to be viewed, in encrypted format.

Output (a collection)

If successful:

1. "OK"
2. Contents of file as text*
3. Actual file name

If failed due to invalid BBM_FileID:

1. "BADID"
2. "Invalid BBM_FileID"
3. Actual file name (this could be rubbish if psBMM_FileID was substantially incorrect)

**Additional Exception Handling*

If file contains unprintable characters the data returned will be preceded by "This may not be an ASCII file - unprintable characters have been replaced by spaces" and as the message advises unprintable characters will be replaced so at least some meaningful information might be visible.

Appendix A – Basic Parameters

All these parameters must be added to the collection created by function BMM_EmptyCollA before reports can be generated (see Appendix B for advanced searching).

List of Parameters

	Key	Type	Notes
1	BMM_params	integer	Fixed value – should be 18
2	BMM_action	integer	0 = no output, used to check parameters are valid 1 = interrogate all matching files and return list of "projects" 5 = As 1 but returns report as HTML 11 Report 15 As 11 but in HTML format 21 Generate report as CSV file 31 Archive - move matching files and (optionally) delete their associated files (e.g. bdp and bdr files) 32 Copy matching files 38 Delete Files 41 Viewer, enables you to step backwards and forwards and view any file from the list of matching files 45 As 41 but for HTML
3	BMM_source	string	The folder to be scanned (sub-folders are ignored)
4	BMM_fileext	string	Used to define which files are to be searched
5	BMM_archive	string	When archiving, where the files are to moved to. *NB. If BMM_makedir = true and this folder does not exist then it will be created during the validation process.*
6	BMM_log	string	The full path and name for your log file. *NB. If BMM_makedir = true and this folder does not exist then it will be created during the validation process.*
7	BMM_CSV	string	When creating CSV output, the full path and name of the file to be generated. *NB. If BMM_makedir = true and this folder does not exist then it will be created during the validation process.*
8	BMM_order	integer	0 = Skip, do not sort (Quickest) 1 = From Address, Line, Find 11 = Date Stamp - Oldest First 12 = Date Stamp - Newest First 21 = Addressee, Status, Date Stamp 22 = Status, Date Stamp
9	BMM_href	string	Only required if creating HTML report with links else leave blank. This is the first part of the link to your web page that displays a file. i.e. "MyFileViewer.aspx?FileId="
10	BMM_maxlines	integer	Optional: 0 or blank if not required Can be used to speed up processing when you

			know the data being searched for is in the first x lines. Recommended default is 100.
11	BMM_sizemin	integer	Optional: 0 or blank if not required Can be used to include / exclude emails with large attachments. Enter size is in kbs.
12	BMM_sizemax	integer	Optional: 0 or blank if not required Can be used to include / exclude emails with large attachments. Enter size is in kbs.
13	BMM_datefrom	string	Optional: blank if not required Format: dd/mm/yyyy HH:MM:SS with HH:MM:SS being optional or can be shortened to HH:MM Special date range commands can be entered: D1 to D99 = Now minus 1 - 99 days to Now H1 to H99 = minus 1 - 99 hours M1 to M99 = minus 1 - 99 minutes
14	BMM_dateto	string	Optional: blank if not required Format: dd/mm/yyyy HH:MM:SS with HH:MM:SS being optional or can be shortened to HH:MM
15	BMM_makedir	boolean	0,1,True or False
16	BMM_delmatch	boolean	0,1,True or False
17	BMM_columns	string	Column numbers delimited by | [pipe] char
18	BMM_license	string	To test your code prior to purchasing a license set to “TRIAL” else enter the license key that you have purchased.
19	BMM_advanced	Integer	No of advanced search sets (see Appendix B)

Appendix B – Advanced Search Parameters (Optional)

For each set of advanced parameters there are three fields. If a field is not being used it still has to entered in the parameters but leave blank .

For Each Advanced Set of Parameters *NB. The number (1-999) needs to be incremented for each set, starting at 1, do not leave any gaps.*	
BMM_advsF1 - 999*	From Address – The from address in the file has to match
BMM_advsL1 - 999*#	Find Line – Enter the whole line you are looking for
BMM_advsT1 - 999*#	Find Text – Each line will be searched for this text on a wild card basis.

Extra features on advanced search:

- * Alternative Text, use "!!!" to display alternative text in reports – you may wish to display a shorter or more meaningful message in the reports, especially when searching for a whole line.

  Example (1) - You want to search for the line "Resetting your Travel-To-The-Moon password in three easy steps" but want to display "Reset Password Emails" in the report.

  ```
  BMM_line = Resetting your Travel-To-The-Moon password in three
  easy steps!!!Reset Password Emails
  ```

- # Multiple Searches within a single set, use "|", pipe. Sometimes you may not want to create separate sets for a multiple search.

  Example (2) - You are searching for joan.smith@somewhere.com but are not certain whether her name is spelt joan or joanne or even just jo.

  ```
  BMM_find = joan.smith@|joanne.smith@|jo.smith@
  ```

NB. When using Alternative Text and Multiple Searches, the alternative text must go at the end.

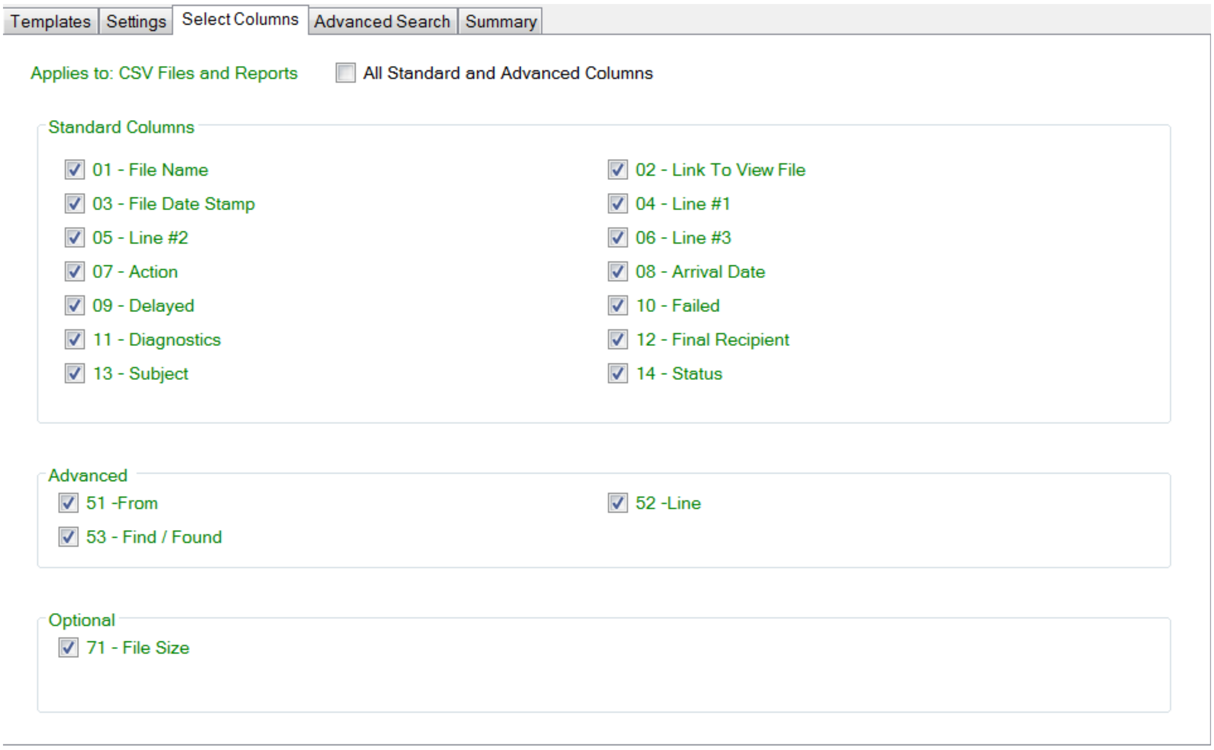
```
BMM_find = joan.smith@|joanne.smith@|jo.smith@!!!Joan Smith
```

Appendix C — Fields that are available in the reports

In the exe these are selected on the Select Columns tab. Basic parameters are set in **BMM_columns**. The required fields are concatenated and delimited by the pipe "|", character.

e.g.

BMM_columns=01|02|05|51|71|



Applies to: CSV Files and Reports

☐ All Standard and Advanced Columns

Standard Columns

☑ 01 - File Name
☑ 02 - Link To View File
☑ 03 - File Date Stamp
☑ 04 - Line #1
☑ 05 - Line #2
☑ 06 - Line #3
☑ 07 - Action
☑ 08 - Arrival Date
☑ 09 - Delayed
☑ 10 - Failed
☑ 11 - Diagnostics
☑ 12 - Final Recipient
☑ 13 - Subject
☑ 14 - Status

Advanced

☑ 51 -From
☑ 52 -Line
☑ 53 - Find / Found

Optional

☑ 71 - File Size

	Heading	Description
1	File Name	Physical File Name
2	File ID	Path + File Name in encrypted format
3	File Date Stamp	Physical file date stamp
4	Line #1	First line of file
5	Line #2	Second line of file
6	Line #3	Third line of file
7	Action	Text appearing after first occurrence of "action:" in the file
8	Arrival-Date	Text appearing after first occurrence of "arrival-date:" in the file
9	Delivery to ... delayed	The line of text after the words "delivery to the following recipients has been delayed"
10	Delivery to ... failed	The line of text after the words "delivery to the following recipients failed" or "unable to deliver message to the following recipients"

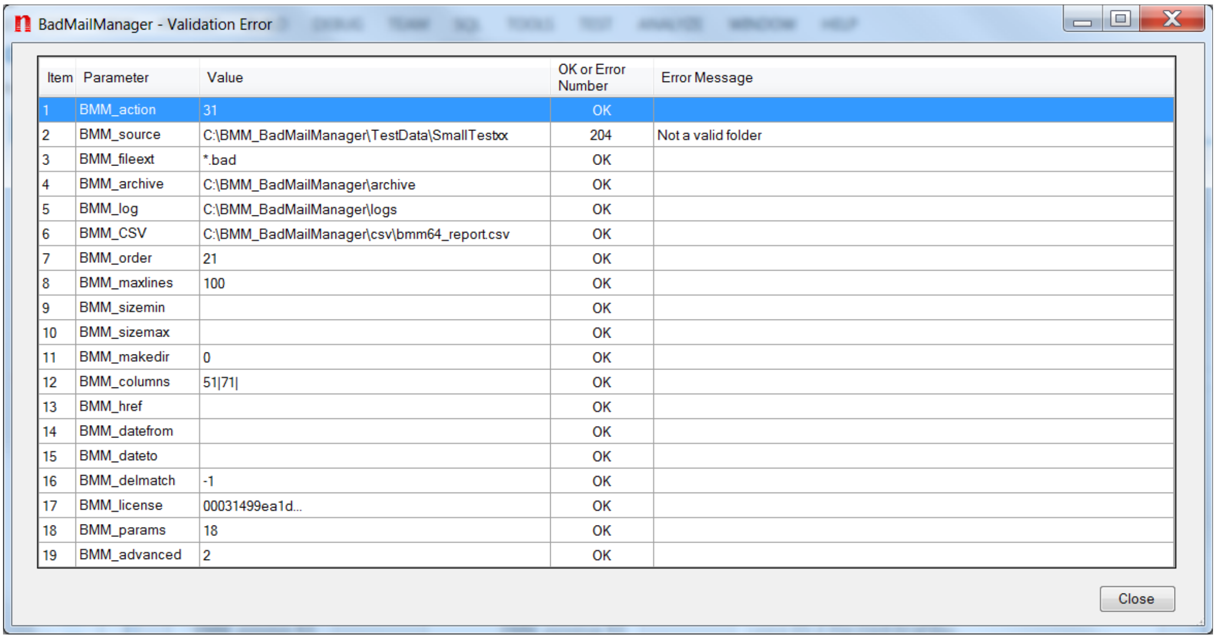
11	Diagnostic Code	Text appearing after first occurrence of “diagnostic-code:” in the file
12	Final Recipient	Text appearing after first occurrence of “final-recipient:” in the file
13	Subject	
14	Status	Text appearing after first occurrence of “status:” in the file
Advanced fields – only contains data if an advanced search criteria was used to select a file		
51	From Address	Filled with the text that this file was selected by
52	Line	- ditto -
53	Search For	- ditto -
Optional		
71	File Size	The length of the file

In some instances the data in a some columns will be the same as in other columns. These fields have been chosen to enable diagnosis of both common issues and those rather odd things that have occasionally appeared in rejected emails.

Appendix D—Parameter Validation

An “INVALID” message is returned when there is a problem with the parameters.

E.g.



Item	Parameter	Value	OK or Error Number	Error Message
1	BMM_action	31	OK	
2	BMM_source	C:\BMM_BadMailManager\TestData\SmallTestxx	204	Not a valid folder
3	BMM_fileext	*.bad	OK	
4	BMM_archive	C:\BMM_BadMailManager\archive	OK	
5	BMM_log	C:\BMM_BadMailManager\logs	OK	
6	BMM_CSV	C:\BMM_BadMailManager\csv\bmm64_report.csv	OK	
7	BMM_order	21	OK	
8	BMM_maxlines	100	OK	
9	BMM_sizemin		OK	
10	BMM_sizemax		OK	
11	BMM_makedir	0	OK	
12	BMM_columns	51|71|	OK	
13	BMM_href		OK	
14	BMM_datefrom		OK	
15	BMM_dateto		OK	
16	BMM_delmatch	-1	OK	
17	BMM_license	00031499ea1d...	OK	
18	BMM_params	18	OK	
19	BMM_advanced	2	OK	

The parameters are returned in a collection of collections. One sub collection for each parameter. Each sub collection having the following fields:

1. Parameters
2. Value
3. Error Number (0 being displayed on the screen as OK – see above)
4. Error Message

Appendix E—List of Common Errors

No.	Function	Description
M1001	BMM64_Mod1_Archive	Invalid or incomplete file name
M1002	BMM64_Mod1_Archive	Unable to move (archive) file name
M1011	BMM64_ArcCopyDelMatching	Unable to archive / copy / delete file name
M1021	BMM64_Mod1_Engine	Error validating parameters
M1022	BMM64_Mod1_Engine	General Error
M1031	BMM64_Mod1_Validate	General Error
M1041	BMM64_ListAsHTML	Error in Engine
M1042	BMM64_ListAsHTML	Error processing Engine output data
M1051	BMM64_ParamsFileGetAndRun	Error Problem validating your parameters file

Appendix F—BMM Parameters File Structure

This is an example of a BMM Parameters file

```
BMM_params=18
BMM_action=41
BMM_source=C:\inetpub\mailroot\badmail
BMM_fileext=*.bad|*.eml
BMM_archive=C:\BMM_BadMailManager\archive
BMM_log=C:\BMM_BadMailManager\logs
BMM_CSV=C:\BMM_BadMailManager\csv\bmm64_report.csv
BMM_order=0
BMM_href=http://yourwebsite.com/bmm64showfile.aspx?bmmid=
BMM_maxlines=100
BMM_sizemin=
BMM_sizemax=
BMM_datefrom=30/01/2017 06:00:00
BMM_dateto=03/02/2017 18:00:00
BMM_makedir=-1
BMM_delmatch=-1
BMM_columns=14|13|12|11|10|09|08|07|06|05|04|03|02|01|53|52|51|
BMM_advanced=2
BMM_advsF1=
BMM_advsL1=2017 Training Courses
BMM_advsT1=ann@anaddress|anne@anaddress
BMM_advsF2=
BMM_advsL2=
BMM_advsT2=julian@anotheraddress
bmmh_date=13/02/2017 20:15:42
bmmh_ver=2.1.1.19
bmmh_lic=00017
bmmh_user=
bmmh_note=This is a sample parameter file
Action is set to Viewer-Browser mode
There are two advanced searches
1) Apparently ann did not get the email with title line of 2017 Training Courses
and the person sending was unsure of whether ann was spelt with or without an e
so trying both on the same search
2) Julian was complaining that he had not received any emails from us that week
bmmh_cs=7268
```

The BMM_… fields are described in Appendices A & B

This example has two advanced searches.

The format for advanced searches is

BMM_advsF1-99= From address
BMM_advsL1-99= Line - can be used to search specific emails
BMM_advsT1-99= Wild card text search, all lines are searched, use for specific emails addresses, types of error etc.

Blank entries have to be included (see ..F1, ..F2, ..L2 above)

The fields:

bmmh_date=	Date file created
bmmh_ver=	Which version of the dll was used to create the file
bmmh_lic=	User’s license number
bmmh_user=	User’s name
bmmh_note=	Notes about this file
bmmh_cs=	A checksum – generated by system but not currently validated so can be set to left as blank

Are used for information only and not used in any processing

Appendix H—HTML Exceptions

Where a function finds matching files the HTML data stream will be an embedded in a table.

If there is an exception one of the following captions will be returned.

No matching files:

```
<caption style='font-size:150%'>No Matching Files</caption>
```

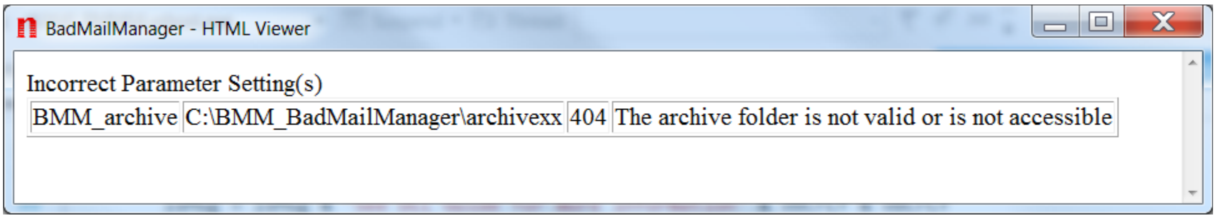
No files with specified extensions:

```
<caption style='font-size:150%'>No Files With Extension of " & BMMColA("BMM_fileext")
& "</caption>
```

Invalid parameters:

```
<caption style='font-size:150%'>Incorrect Parameter Setting(s)</caption>
```

With Parameter Number, Name, Value, Error Number, Error Description in table e.g.



Incorrect Parameter Setting(s)

BMM_archive	C:\BMM_BadMailManager\archivexx	404	The archive folder is not valid or is not accessible

Errors:

```
<caption style='font-size:150%'>Unable to run report [Details: Error XXXXXX]</caption>
```

NB. There are no <html> </html> tags included to make it easier to insert the HTML data in your page.

END