Error: Macro TranslatedPages(None) failed

```
expected string or buffer
```

ARexx Function List

1. ADDRDELETE
2. ADDREDIT
3. ADDRFIND
4. ADDRGOTO
5. ADDRINFO
6. ADDRLOAD
7. ADDRNEW
8. ADDRRESOLVE
9. ADDRSAVE
10. APPBUSY
11. APPNOBUSY
12. FINDMAIL
13. FLUSHINDEXES
14. FOLDERINFO
15. GETCONFIGINFO
16. GETFOLDERINFO
17. GETMAILINFO
18. GETSELECTED
19. GETURL
20. HELP
21. HIDE
22. INFO
23. ISONLINE
24. LISTFREEZE
25. LISTSELECT
26. LISTUNFREEZE
27. MAILARCHIVE
28. MAILBOUNCE
29. MAILCHANGESUBJECT
30. MAILCHECK
31. MAILCOPY
32. MAILDELETE
33. MAILEDIT
34. MAILEXPORT
35. MAILFILTER
36. MAILFORWARD
37. MAILIMPORT
38. MAILINFO
39. MAILMOVE
40. MAILREAD
41. MAILREPLY
42. MAILSEND
43. MAILSENDALL
44. MAILSTATUS
45. MAILUPDATE
46. MAILWRITE
47. NEWMAILFILE
48. QUIT

49. READCLOSE
50. READINFO
51. READPRINT
52. READSAVE
53. REQUEST
54. REQUESTFILE
55. REQUESTFOLDER
56. REQUESTSTRING
57. RESTART
58. SCREENTOBACK
59. SCREENTOFRONT
60. SETFLAG
61. SETFOLDER
62. SETMAIL
63. SETMAILFILE
64. SHOW
65. USERINFO
66. WRITEATTACH
67. WRITEBCC
68. WRITECC
69. WRITEEDITOR
70. WRITEFROM
71. WRITEIDENTITY
72. WRITELETTER
73. WRITEMAILTO
74. WRITEOPTIONS
75. WRITEQUEUE
76. WRITEREPLYTO
77. WRITESEND
78. WRITESUBJECT
79. WRITETO

Reference - ARexx interface

With the help of ARexx scripts, you can add new functions to YAM or let it do things automatically.
YAM offers a set of commands (listed in this documentation both by name ? and by function ?) which can be called through the **YAM** ARexx port. They are explained along the following subchapters using the following format:

NAME
: The name of the command, a short description of what it does and the YAM version where the command was implemented.

TEMPLATE
: Arguments and options accepted by the command. The template uses special characters that indicate the particular type of argument expected, following the AmigaDOS template style:

`/A` The parameter is compulsory
`/K` The parameter must be preceeded by the keyword
`/N` Numerical argument or result
`/M` Argument or result is a list of zero or more elements
`/S` The parameter works as a switch; the switch is in use when the parameter is given

FUNCTION

Describes what the command will do.

INPUTS

Describes in detail the parameters accepted by the command. Be careful when passing arguments containing spaces; for instance,

```
sub = 'Hello World'
'WRITESUBJECT' sub
```

won't work. It must be written as

```
'WRITESUBJECT "'sub'"'
```

or

```
'WRITESUBJECT "Hello World"'
```

Please note that because of the internal use of the ReadArgs() function, the ARexx Host requires to escape certain special characters like a newline (0x0a) or escape character (0x1b) if you want to have it included in your final string or otherwise it is stripped by the ReadArgs() function.

This means that you have to use the following escape sequences in your provided strings:

`*N` substitutes to 0x0a (newline \n)
`*E` substitutes to 0x1b (escape)
`!**` substitutes to *
`*` substitutes to "

For example the following command would write a string to a texteditor containing a newline:

```
'WRITEEDITOR "TEXT Hello Joe,*N I would like to meet you."'
```

If a parameter represents a file name you should keep in mind that file names on AmigaOS might contain spaces or other special characters which will cause ARexx to treat the name as multiple words instead of a single word. Hence a file name should **always** be surrounded by quotes, no matter if it contains spaces or not. Scripts can very easily be tested for this case by letting them handle a file in the Amiga RAM disk, i.e. "RAM Disk:/foo/bar".

RETURNS

Returned info to be expected from the command. Commands may return results in three different kinds; let's look at these examples which use the FOLDERINFO command:

```
FOLDERINFO
-> RESULT = "0 Incoming incoming 10 2 4 23030 1"

FOLDERINFO VAR fi
-> fi = "0 Incoming incoming 10 2 4 23030 1"

FOLDERINFO STEM fi.
-> fi.number = 0
     fi.name = "Incoming"
     fi.path = "incoming"
     fi.total = 10
     fi.new = 2
     fi.unread = 4
     fi.size = 23030
```

```
     fi.type = 1
```

Another example which returns a result of type `/M`:

```
ADDRFIND STEM found. "Marcel Beck" NAMEONLY
 -> found.alias.count = 2
      found.alias.0 = "Mars"
      found.alias.1 = "mbe"
```

WARNING
: Any sort of vital information you should be aware of when using this command.

NOTES
: Various notes about the command.

EXAMPLE
: A fragment of ARexx code to illustrate the usage of the command.

BUGS
: Problems that are known or have already been fixed with this command.

SEE ALSO
: Links to other related commands.

- Alphabetical list of all ARexx commands?
- List of all the ARexx commands sorted by function?