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
*E substitutes to 0x1b
** 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."'
```

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.