

# **MCC-Install\_Developer**

Dipl.-Inform. Kai Hofmann

Copyright © CopyrightÂ©1999 Dipl.-Inform Kai Hofmann. All rights reserved.

COLLABORATORS

	TITLE : MCC-Install_Developer		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Dipl.-Inform. Kai Hofmann	August 25, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>MCC-Install_Developer</b>	<b>1</b>
1.1	MCC-Install_Developer . . . . .	1
1.2	Custom functions . . . . .	1
1.3	P_CustomExists . . . . .	1
1.4	P_CustomInstall . . . . .	2
1.5	P_CustomUninstall . . . . .	4
1.6	Script internal functions . . . . .	6
1.7	P_BitToLanguage . . . . .	6
1.8	P_GetVersion . . . . .	7
1.9	P_CompareVersions . . . . .	7
1.10	P_GetOSVersionStr . . . . .	8
1.11	P_GetOSVersion . . . . .	8
1.12	P_GetCPUFPU . . . . .	9
1.13	P_AskLanguages . . . . .	10
1.14	P_AskDocFormats . . . . .	10
1.15	P_AskProgLangs . . . . .	11
1.16	P_DelEmptyDir . . . . .	11
1.17	P_DelDirRecursive . . . . .	12
1.18	P_DeleteLanguages . . . . .	12
1.19	P_MkdirInfo . . . . .	13
1.20	P_CpFile . . . . .	13
1.21	P_GetMuiVersionStr . . . . .	14
1.22	P_GetMUIVersion . . . . .	14
1.23	P_AskCPUFPU . . . . .	15
1.24	P_AskParts . . . . .	16
1.25	P_CheckAvailableCPUs . . . . .	16
1.26	P_CheckExists . . . . .	17
1.27	P_Install . . . . .	18
1.28	P_Uninstall . . . . .	20
1.29	MCC-Install main . . . . .	22

1.30 Script internal variables . . . . . 23

1.31 Languages bitmask . . . . . 23

1.32 CPU bitmask . . . . . 24

1.33 FPU bitmask . . . . . 24

1.34 Document type bitmask . . . . . 24

1.35 Programming languages bitmask . . . . . 24

1.36 MCC/MCP cpu file extension string . . . . . 25

1.37 Archiv parts bitmask . . . . . 25

1.38 Useful text strings . . . . . 25

## Chapter 1

# MCC-Install\_Developer

### 1.1 MCC-Install\_Developer

Table of Contents

**Custom functions** - Custom functions

**Internal functions** - Script internal functions

**Internal variables** - Script internal variables

**Useful text strings** - Strings that might be useful for custom functions

If you have AmigaDOS® 2.0 or 2.1, you will have to scroll to the right in order to read the text. AmigaGuide® for these versions of AmigaDOS® is unable to wordwrap the text.

### 1.2 Custom functions

Custom functions

-----

**P\_CustomExists** - Custom procedure to check existens

**P\_CustomInstall** - Custom procedure for install

**P\_CustomUninstall** - Custom procedure for uninstall

### 1.3 P\_CustomExists

P\_CustomExists

=====

Available variables

-----

(See MCC-Install main)

#copyright

#min\_os\_version

#min\_os\_revision  
 #min\_mui\_version  
 #min\_mui\_revision  
 #language\_mcc  
 #language\_mcp  
 #prop  
 #language\_default  
 various language strings  
 (other)  
 #os\_ver (See P\_GetOsVersion)  
 #os\_rev  
 #os\_str  
 #mui\_ver (See P\_GetMUIVersion)  
 #mui\_rev  
 #mui\_str  
 #src-dir (directory of the archiv)  
 #dest-dir (where MUI is installed)  
 #dir (same as #dest-dir)  
 All P\_CheckExists variables for things that are already installed(!)  
 (Installer internals)  
 (to be used as temps)  
 #i  
 Function

-----

Should initialize custom chk\_ and old\_ variables for things that exists within the archiv/mui installation - if required by P\_CustomInstall, P\_CustomUninstall

Problems

-----

Maybe P\_CustomExists should be called twice in the same way as P\_CheckExists. The Problem is, that there is no way to copy chk\_ to old\_ after calling it the first time - so P\_CustomExists has to work with #src-dir and #dest-dir instead of #dir

## 1.4 P\_CustomInstall

P\_CustomInstall

=====

Available variables

-----

chk\_ = within the archiv

old\_ = already installed

---

#update (0/1 new or already installed mcc)  
#src-dir (directory structur of the archiv)  
#dest-dir (directory structur of MUI)  
#abstract (0/1 abstract mcc)  
#install\_mcc\_catalogs (MCC catalogs to install)  
#install\_mcp\_catalogs (MCP catalogs to install)  
#install\_docs (Doc. types to install)  
#install\_asciidocs (ASCII docs to install)  
#install\_agdocs (AmigaGuide docs to install)  
#install\_htmldocs (HTML docs to install)  
#install\_demo (0/1 install demo)  
#install\_proglang (Programming languages to install)  
#install\_source (0/1 install source code)  
#install\_images (0/1 install images)  
#os\_ver (OS Version number)  
#os\_rev (OS revision number)  
#os\_str (OS Release string)  
#mui\_ver (MUI Version number)  
#mui\_rev (MUI Revision number)  
#mui\_str (MUI Release version string)  
#cpu (cpu bitmask)  
#cpu\_str (cpu name as string)  
#mcc\_cpu\_file\_extension (Mcc file extension for choosed CPU)  
#mcp\_cpu\_file\_extension (Mcp file extension for choosed CPU)  
#mcc\_cpus (available mcc/cpu bitmask)  
#mcp\_cpus (available mcp/cpu bitmask)  
#mcc\_fpus (available mcc/fpu bitmask)  
#mcp\_fpus (available mcp/fpu bitmask)  
#mcc\_cpu\_counter (number of available mcc cpu's)  
#mcp\_cpu\_counter (number of available mcp cpu's)  
#chk\_mcp (0/1 custom class exists)  
#chk\_demo (0/1 demo exists)  
#chk\_mcc\_catalogs (language bitmask for mcc catalogs)  
#chk\_mcp\_catalogs (language bitmask for mcp catalogs)  
#chk\_docs (bitmask for available doc types)  
#chk\_asciidocs (language bitmask for ASCII docs)  
#chk\_agdocs (language bitmask for AmigaGuide docs)  
#chk\_htmldocs (language bitmask for HTML docs)  
#chk\_proglang (bitmask for completly available programming languages)

---



#chk\_progdirs (bitmask for available programming lanuages directories)  
 #chk\_autodoc (0/1 Autodocs available)  
 #chk\_source (0/1 class source code available)  
 #chk\_images (0/1 Images available)  
 #chk\_existsmask (bitmask for existing parts)  
 #old\_mcp (0/1 mcp exists)  
 #old\_demo (0/1 demo exists)  
 #old\_mcc\_catalogs (language bitmask for mcc catalogs)  
 #old\_mcp\_catalogs (language bitmask for mcp catalogs)  
 #old\_docs (bitmask for available doc types)  
 #old\_asciidocs (language bitmask for ASCII docs)  
 #old\_agdocs (language bitmask for AmigaGuide docs)  
 #old\_proglang (bitmask for completly available programming languages)  
 #old\_progdirs (bitmask for available programming lanuages directories)  
 #old\_autodoc (0/1 Autodocs available)  
 Temporary variables you should use:  
 #prompt (various prompt texts)  
 #defaultmask (various default values)  
 #langmask (various language masks)  
 #infos (0/1 copy .info files)  
 #dir (various dirs)  
 #file (various files)  
 #pattern (0/1 copy pattern)

Function

-----

Installs additional parts, that are not covered in the regular MCC-Install process.

## 1.5 P\_CustomUninstall

P\_CustomUninstall

=====

Available variables

-----

#dest-dir (directory structur where MUI ist installed)  
 #all (Uninstall all parts?)  
 #parts (see P\_AskParts)  
 #uninstall\_mcc\_catalogs (MCC catalogs to uninstalled)  
 #uninstall\_mcp\_catalogs (MCP catalogs to uninstalled)  
 #uninstall\_asciidocs (ASCII docs to uninstall)

#uninstall\_agdocs (AmigaGuide docs to uninstall)  
#uninstall\_htmldocs (HTML docs to uninstall)  
#uninstall\_proglang (Programming languages to uninstall)  
#old\_mcc (0/1 custom class exists)  
#old\_mcp (0/1 mcp exists)  
#old\_demo (0/1 demo exists)  
#old\_mcc\_catalogs (language bitmask for mcc catalogs)  
#old\_mcp\_catalogs (language bitmask for mcp catalogs)  
#old\_catalogs (language bitmask #chk\_mcc\_catalogs OR #chk\_mcp\_catalogs)  
#old\_docs (bitmask for available doc types)  
#old\_asciidocs (language bitmask for ASCII docs)  
#old\_htmldocs (language bitmask for HTML docs)  
#old\_agdocs (language bitmask for AmigaGuide docs)  
#old\_proglang (bitmask for completely available programming languages)  
#old\_autodoc (0/1 Autodocs available)  
#old\_c\_h (0/1 C header file available)  
#old\_c\_example (0/1 C example available)  
#old\_asm\_i (0/1 Assembler include file available)  
#old\_asm\_example (0/1 Assembler example available)  
#old\_e\_e (0/1 Amiga E module available)  
#old\_e\_example (0/1 Amiga E example available)  
#old\_modula\_def (0/1 Modula-2 Def available)  
#old\_modula\_mod (0/1 Modula-2 Mod available)  
#old\_modula\_demo (0/1 Modula-2 Demo available)  
#old\_oberon\_mod (0/1 Oberon Mod available)  
#old\_oberon\_example (0/1 Oberon example available)  
#old\_rexx (0/1 ARexx include available)  
#old\_rexx\_example (0/1 ARexx example available)  
#old\_cluster\_def (0/1 Cluster Def available)  
#old\_cluster\_mod (0/1 Cluster Mod available)  
#old\_cluster\_demo (0/1 Cluster example available)  
#old\_bb2 (0/1 Blitz Basic 2 include available)  
#old\_bb2\_example (0/1 Blitz basic 2 example available)  
#old\_muibuilder3 (0/1 MUIBuilder 3 definition available)  
#old\_muibuilder3\_behaviour (0/1 MUIBuilder 3 behaviour available)  
#old\_source (0/1 class source code available)  
#old\_existsmask (bitmask for existing parts)  
Temporary variables you should use:  
#prompt (for prompts)

---

#defaultmask (for defaultmasks)

#langmask (for language masks)

Function

-----

Uninstalls additional parts, that are not covered in the regular MCC-Install process.

## 1.6 Script internal functions

Script internal functions

-----

**P\_BitToLanguage** - Language bitmap to language string

**P\_GetVersion** - Get version for a file

**P\_CompareVersions** - Compare two versions

**P\_GetOSVersionStr** - Get OS version string

**P\_GetOSVersion** - Get OS version

**P\_GetCPUFPU** - Get available cpu/fpu

**P\_AskLanguages** - Ask user for languages

**P\_AskDocFormats** - Ask user for document formats

**P\_AskProgLangs** - Ask user for programming languages

**P\_DelEmptyDir** - Delete empty directory

**P\_DelDirRecursive** - Delete a directory recursiv

**P\_DeleteLanguages** - Delete languages directories

**P\_MkDirInfo** - Create new direktory

**P\_CpFile** - Copy file

**P\_GetMuiVersionStr** - Get MUI version string

**P\_GetMUIVersion** - Get MUI version

**P\_AskCPUFPU** - Ask user for cpu/fpu

**P\_AskParts** - Ask user for parts

**P\_CheckAvailableCPUs** - Check available cpu binaries

**P\_CheckExists** - Check for existing parts

**P\_Install** - Install mcc

**P\_Uninstall** - Uninstall mcc

**MCC-Install main** - MCC-Install main part

## 1.7 P\_BitToLanguage

P\_BitToLanguage

=====

Parameter

-----

#langmask (bitmask see variables)

Temps

-----

#i (counter)

#bitmask (working copy of #langmask)

Results

-----

#langmask (mask without choosed language)

#lang (language name string)

Function

-----

Returns the next language name string from #langmask - and removes this language from #langmask (for looping through all langs)

## 1.8 P\_GetVersion

P\_GetVersion

=====

Parameter

-----

#file (Filename for which you want to know the version)

Temps

-----

#vernum (Installer version/revision-num format)

Results

-----

#ver (Version)

#rev (Revision)

Function

-----

Returns the version and revision for a file

## 1.9 P\_CompareVersions

P\_CompareVersions

=====

Parameter

-----

#ver (First version number)  
 #rev (First revision number)  
 #ver2 (Second version number)  
 #rev2 (Second revision number)

Temps

-----

Results

-----

#result

-1 : #ver.#rev < #ver2.#rev2

0 : #ver.#rev = #ver2.#rev2

1 : #ver.#rev > #ver2.#rev2

Function

-----

Compares two version/revision pairs

## 1.10 P\_GetOSVersionStr

P\_GetOSVersionStr

=====

Parameter

-----

#ver (OS Version number (30-40))

#rev (OS Revision Number [currently unused])

Temps

-----

#i (counter)

Results

-----

#str (OS Version String)

Function

-----

Returns the OS Release name string from the OS version number

## 1.11 P\_GetOSVersion

P\_GetOSVersion

=====

Parameter

-----

Temps

-----

#file ("libs:version.library")

#ver (see Used Funcs)

#rev (see Used Funcs)

#vernum (see Used Funcs)

#i (see Used Funcs)

#str (see Used Funcs)

Results

-----

#os\_ver (OS Version number)

#os\_rev (OS revision number)

#os\_str (OS Release string)

Function

-----

Returns the OS version/revision number and the release string

Used Functions

-----

P\_GetVersion, P\_GetOSVersionStr

## 1.12 P\_GetCPUFPU

P\_GetCPUFPU

=====

Parameter

-----

Temps

-----

Results

-----

#cpu (available CPU)

-1 = unknown

0 = 68000

1 = 68010

2 = 68020

3 = 68030

4 = 68040

5 = 68060

#cpu\_str (available CPU as string: 68000,68010,68020,68030,68040,68060)

#fpu (available FPU)

-1 = unknown

0 = NOFPU

1 = 68881

2 = 68882

3 = FPU40

#fpu\_str (available FPU as string: NOFPU,68881,68882,FPU40)

Function

-----

Returns the available CPU on the current system

## 1.13 P\_AskLanguages

P\_AskLanguages

=====

Parameter

-----

#prompt (Prompt to display i.e.: choose languages for catalogs)

#langmask (Language mask of available languages [see Variables])

#defaultmask (Default language mask [pre selected languages])

Temps

-----

#bitmask (working copy of #langmask)

Results

-----

#langmask (Mask of selected languages [see Variables])

Function

-----

Ask the user to select languages

## 1.14 P\_AskDocFormats

P\_AskDocFormats

=====

Parameter

-----

#prompt (Prompt text i.e. which type of doc should be installed)

#docmask (Available doc formats bitmask)

#defaultmask (Preselected formats)

Temps

-----

#bitmask (working copy of #docmask)

Results

-----

#docmask (Formats the user has choosen)

Function

-----

Ask the user for the doc-type to install (ACSII, AmigaGuide, HTML)

## 1.15 P\_AskProgLangs

P\_AskProgLangs

=====

Parameter

-----

#prompt (Prompt to ask the user)

#plangmask (Available programming languages)

#defaultmask (Pre selected programming languages)

Temps

-----

#bitmask (Working copy of #plangmask)

Results

-----

#plangmask (User selected programming languages)

Function

-----

Ask the user/developer for a programming language

## 1.16 P\_DelEmptyDir

P\_DelEmptyDir

=====

Parameter

-----

#dir (Directory to delete)

Temps

-----



#i (counter / number of deleted directories)

Results

-----

Function

-----

Deletes only empty directories including their icons

## 1.17 P\_DelDirRecursive

P\_DelDirRecursive

=====

Parameter

-----

#dir (Directory to delete)

Temps

-----

Results

-----

Function

-----

Deletes a directory recursively including icons

Used Functions

-----

P\_DelDirRecursive

## 1.18 P\_DeleteLanguages

P\_DeleteLanguages

=====

Parameter

-----

#dir (directory which contains language dirs)

#file (file to delete from languages dir)

#langmask (languages mask for which #file should be deleted)

Temps

-----

#i (counter)

#lang (temp language name string)

Results

-----

#langmask (empty langmask)

Function

-----

Deletes a (catalog) file from all selected language directories

Used Functions

-----

P\_BitToLanguage

## 1.19 P\_MkDirInfo

P\_MkDirInfo

=====

Parameter

-----

#dir (dir name to create)

#infos (create icon file [0/1])

#dest-dir (already existing path within which #dir should be created)

Temps

-----

Results

-----

Function

-----

Creates a new directory including icon file if required

## 1.20 P\_CpFile

P\_CpFile

=====

Parameter

-----

#dir (directory to copy)

#file (filepattern for copyoperation)

#infos (copy icons [0/1])

#dest-dir (destination directory)

#src-dir (directory with sources)

#pattern (copy single file or filepattern [0/1])

Temps

---

-----

Results

-----

Function

-----

Copies a single file or multiple via patterns including icons if required

## 1.21 P\_GetMuiVersionStr

P\_GetMuiVersionStr

=====

Parameter

-----

#ver (MUI Version number [i.e. 38])

#rev (MUI Revision number)

Temps

-----

#i (counter)

Results

-----

#str (MUI release version string [i.e. "3.8"])

Function

-----

Returns the MUI release version string from the library version number

## 1.22 P\_GetMUIVersion

P\_GetMUIVersion

=====

Parameter

-----

Temps

-----

#file (MUI library/preferences filename)

#ver (MUI version num)

#rev (MUI revision num)

#vernum (MUI version (installer format))

#i (counter)

Results

-----

#mui\_ver (MUI Version number)  
 #mui\_rev (MUI Revision number)  
 #mui\_str (MUI Release version string)  
 Function

-----

Returns the MUI Version/revision number as well as the release string

## 1.23 P\_AskCPUFPU

P\_AskCPUFPU

=====

Parameter

-----

#src-dir (directory with MUI files)

Temps

-----

#i (counter)  
 #file (see used funcs)  
 #bitmask (see used funcs)  
 #filetype (see used funcs)  
 #f (see used funcs)  
 #mcc\_cpus (available binary versions)  
 #mcp\_cpus (available binary versions)  
 #mcc\_fpus (available binary versions)  
 #mcp\_fpus (available binary versions)  
 #dir (directory for P\_CheckAvailableCPUs)  
 #defaultcpu (best fit cpu)  
 #mcc\_cpu\_counter (number of available mcc cpu's)  
 #mcp\_cpu\_counter (number of available mcp cpu's)  
 #fpu (see used funcs)  
 #fpu\_str (see used funcs)  
 #cpu2 (next lower usable cpu)  
 #fpu2 (next lower usable fpu)  
 Results  
 -----  
 #cpu (cpu bitmask)  
 #cpu\_str (cpu name as string)  
 #mcc\_cpu\_file\_extension (Mcc file extension for choosed CPU)

#mcp\_cpu\_file\_extension (Mcp file extension for choosed CPU)

#src-dir (directory with MUI files)

Function

-----

Let the user choose from available and usable cpu/fpu and return the best fit for mcp (if only a lower cpu is available)

Used Functions

-----

P\_CheckAvailableCPUs, P\_GetCPUFPU

## 1.24 P\_AskParts

P\_AskParts

=====

Parameter

-----

#prompt (ask the user for install/deinstall/update)

#defaultmask (bitmask - see #parts variable - default selection)

#existsmask (bitmask - see #parts variable - existing things)

Temps

-----

Results

-----

#parts (bitmask - see variables)

Function

-----

Ask the user for parts to install/deinstall parts (only available part)

## 1.25 P\_CheckAvailableCPUs

P\_CheckAvailableCPUs

=====

Parameter

-----

#dir (directory with custom class)

Temps

-----

#i (counter)

#file (temporary filename)

#bitmask (temp cpu/fpu bitmask)

#filetype (temp filetype)

#f (temp pattern match)

Results

-----

#mcc\_cpus (available mcc/cpu bitmask)

#mcp\_cpus (available mcp/cpu bitmask)

#mcc\_fpus (available mcc/fpu bitmask)

#mcp\_fpus (available mcp/fpu bitmask)

#mcc\_cpu\_counter (number of available mcc cpu's)

#mcp\_cpu\_counter (number of available mcp cpu's)

Function

-----

Returns the available mcc/mcp cpu's/fpu's

## 1.26 P\_CheckExists

P\_CheckExists

=====

Parameter

-----

#dir (directory structur to check [archiv/mui installation])

Temps

-----

#i (counter)

#lang (language name string)

Results

-----

#chk\_mcc (0/1 custom class exists)

#chk\_mcp (0/1 mcp exists)

#chk\_demo (0/1 demo exists)

#chk\_mcc\_catalogs (language bitmask for mcc catalogs)

#chk\_mcp\_catalogs (language bitmask for mcp catalogs)

#chk\_catalogs (language bitmask #chk\_mcc\_catalogs OR #chk\_mcp\_catalogs)

#chk\_agdocs (language bitmask for AmigaGuide docs)

#chk\_asciidocs (language bitmask for ASCII docs)

#chk\_htmldocs (language bitmask for HTML docs)

#chk\_docs (bitmask for available doc types)

#chk\_progdirs (bitmask for available programming lanuages directories)

#chk\_proglang (bitmask for completly available programming languages)

```

#chk_autodoc (0/1 Autodocs available)
#chk_c_h (0/1 C header file available)
#chk_c_example (0/1 C example available)
#chk_asm_i (0/1 Assembler include file available)
#chk_asm_example (0/1 Assembler example available)
#chk_e_e (0/1 Amiga E modul available)
#chk_e_example (0/1 Amiga E example available)
#chk_modula_def (0/1 Modula-2 Def available)
#chk_modula_mod (0/1 Modula-2 Mod available)
#chk_modula_demo (0/1 Modula-2 Demo available)
#chk_oberon_mod (0/1 Oberon Mod available)
#chk_oberon_example (0/1 Oberon example available)
#chk_rexx (0/1 ARexx include available)
#chk_rexx_example (0/1 ARexx example available)
#chk_cluster_def (0/1 Cluster Def available)
#chk_cluster_mod (0/1 Cluster Mod available)
#chk_cluster_demo (0/1 Cluster example available)
#chk_bb2 (0/1 Blitz Basic 2 include available)
#chk_bb2_example (0/1 Blitz basic 2 example available)
#chk_muibuilder3 (0/1 MUIBuilder 3 definition available)
#chk_muibuilder3_behaviour (0/1 MUIBuilder 3 behaviour available)
#chk_source (0/1 class source code available)
#chk_images (0/1 Images available)
#chk_existsmask (bitmask for existing parts)
Function
-----

```

Check for existing files within archiv or mui installation

## 1.27 P\_Install

P\_Install

=====

Parameter

-----

chk\_ = within the archiv

old\_ = already installed

#chk\_mcp (0/1 custom class exists)

#chk\_demo (0/1 demo exists)

#chk\_mcc\_catalogs (language bitmask for mcc catalogs)

#chk\_mcp\_catalogs (language bitmask for mcp catalogs)  
#chk\_docs (bitmask for available doc types)  
#chk\_asciidocs (language bitmask for ASCII docs)  
#chk\_agdocs (language bitmask for AmigaGuide docs)  
#chk\_htmldocs (language bitmask for HTML docs)  
#chk\_proglang (bitmask for completely available programming languages)  
#chk\_progdirts (bitmask for available programming languages directories)  
#chk\_autodoc (0/1 Autodocs available)  
#chk\_source (0/1 class source code available)  
#chk\_images (0/1 Images available)  
#chk\_existsmask (bitmask for existing parts)  
#old\_mcp (0/1 mcp exists)  
#old\_demo (0/1 demo exists)  
#old\_mcc\_catalogs (language bitmask for mcc catalogs)  
#old\_mcp\_catalogs (language bitmask for mcp catalogs)  
#old\_docs (bitmask for available doc types)  
#old\_asciidocs (language bitmask for ASCII docs)  
#old\_agdocs (language bitmask for AmigaGuide docs)  
#old\_proglang (bitmask for completely available programming languages)  
#old\_progdirts (bitmask for available programming languages directories)  
#old\_autodoc (0/1 Autodocs available)  
#update (0/1 new or already installed mcc)  
#src-dir (directory structure of the archive)  
#dest-dir (directory structure of MUI)  
#abstract (0/1 abstract mcc)  
Temps  
-----  
#prompt (various prompt texts)  
#existsmask (bitmask for already existing parts)  
#defaultmask (various default values)  
#langmask (various language masks)  
#docmask (Document types)  
#infos (0/1 copy .info files)  
#dir (various dirs)  
#file (various files)  
#pattern (0/1 copy pattern)  
#mcc\_cpu\_counter (Nr. of available mcc CPUs)  
#install\_mcc\_catalogs (MCC catalogs to install)  
#install\_mcp\_catalogs (MCP catalogs to install)

---



```
#install_docs (Doc. types to install)
#install_asciidocs (ASCII docs to install)
#install_agdocs (AmigaGuide docs to install)
#install_htmldocs (HTML docs to install)
#install_demo (0/1 install demo)
#install_proglang (Programming languages to install)
#install_source (0/1 install source code)
#install_images (0/1 install images)
```

Results

-----

Function

-----

Installs/Updates the MCC or parts of it

Used Functions

-----

P\_AskParts, P\_AskCPUFPU, P\_AskLanguages, P\_AskDocFormats, P\_AskProgLangs, P\_MkDirInfo, P\_BitToLanguage, P\_CpFile, P\_CustomInstall

## 1.28 P\_Uninstall

P\_Uninstall

=====

Parameter

-----

```
#dest-dir (directory structur where MUI ist installed)
#old_mcc (0/1 custom class exists)
#old_mcp (0/1 mcp exists)
#old_demo (0/1 demo exists)
#old_mcc_catalogs (language bitmask for mcc catalogs)
#old_mcp_catalogs (language bitmask for mcp catalogs)
#old_catalogs (language bitmask #chk_mcc_catalogs OR #chk_mcp_catalogs)
#old_docs (bitmask for available doc types)
#old_asciidocs (language bitmask for ASCII docs)
#old_htmldocs (language bitmask for HTML docs)
#old_agdocs (language bitmask for AmigaGuide docs)
#old_proglang (bitmask for completly available programming languages)
#old_autodoc (0/1 Autodocs available)
#old_c_h (0/1 C header file available)
#old_c_example (0/1 C example available)
```

#old\_asm\_i (0/1 Assembler include file available)  
#old\_asm\_example (0/1 Assembler example available)  
#old\_e\_e (0/1 Amiga E module available)  
#old\_e\_example (0/1 Amiga E example available)  
#old\_modula\_def (0/1 Modula-2 Def available)  
#old\_modula\_mod (0/1 Modula-2 Mod available)  
#old\_modula\_demo (0/1 Modula-2 Demo available)  
#old\_oberon\_mod (0/1 Oberon Mod available)  
#old\_oberon\_example (0/1 Oberon example available)  
#old\_rexx (0/1 ARexx include available)  
#old\_rexx\_example (0/1 ARexx example available)  
#old\_cluster\_def (0/1 Cluster Def available)  
#old\_cluster\_mod (0/1 Cluster Mod available)  
#old\_cluster\_demo (0/1 Cluster example available)  
#old\_bb2 (0/1 Blitz Basic 2 include available)  
#old\_bb2\_example (0/1 Blitz basic 2 example available)  
#old\_muibuilder3 (0/1 MUIBuilder 3 definition available)  
#old\_muibuilder3\_behaviour (0/1 MUIBuilder 3 behaviour available)  
#old\_source (0/1 class source code available)  
#old\_existsmask (bitmask for existing parts)  
Temps  
-----  
#existsmask (bitmask for existing parts)  
#all (Uninstall all parts?)  
#prompt (see P\_AskParts, P\_AskLanguages, P\_AskDocFormats, P\_AskProgLangs)  
#defaultmask (see P\_AskParts, P\_AskLanguages, P\_AskDocFormats, P\_AskProgLangs)  
#parts (see P\_AskParts)  
#langmask (see P\_AskLanguages)  
#uninstall\_mcc\_catalogs (MCC catalogs to uninstall)  
#uninstall\_mcp\_catalogs (MCP catalogs to uninstall)  
#docmask (see P\_AskDocFormats)  
#uninstall\_asciidocs (ASCII docs to uninstall)  
#uninstall\_agdocs (AmigaGuide docs to uninstall)  
#uninstall\_htmldocs (HTML docs to uninstall)  
#plangmask (see P\_AskProgLangs)  
#uninstall\_proglang (Programming languages to uninstall)  
Results  
-----  
Function

---

-----

Uninstalls the MCC or parts of it

Used Functions

-----

P\_AskParts, P\_AskLanguages, P\_AskDocFormats, P\_AskProgLangs, P\_DeleteLanguages, P\_BitToLanguage, P\_DelEmptyDir, P\_DelDirRecursive, P\_CustomUninstall

## 1.29 MCC-Install main

MCC-Install main

=====

Parameter

-----

#copyright (copyright for the mcc)

#min\_os\_version (minimum required OS version)

#min\_os\_revision (minimum required OS revision)

#min\_mui\_version (minimum required MUI version)

#min\_mui\_revision (minimum required MUI revision)

#language\_mcc (mcc buildin languages)

#language\_mcp (mcp buildin languages)

Temps

-----

#prop (Use proportional font with newer installer)

#i (counter)

#language (language name string of user environment)

#language\_default (language bitmask of user environment)

#abstract (0/1 abstract mcc)

#update (0/1 new or already installed)

Various installation strings for user prompting.

All variables from P\_CheckExists prefixed with old\_ instead of chk\_ : old\_ already installed, chk\_ available within the archiv.

See above functions

Results

-----

Function

-----

Installs/uninstalls MUI custom classes

Used Functions

-----

P\_GetOSVersion, P\_CompareVersions, P\_GetMUIVersion, P\_GetMUIVersionStr, P\_CheckExists, P\_CustomExists, P\_GetVersion, P\_Install, P\_Uninstall

## 1.30 Script internal variables

Script internal variables

-----

**#languages** - Bitmask for languages

**#cpu** - Bitmask for cpus

**#fpu** - Bitmask for fpus

**#docmask** - Bitmask for document types

**#plangmask** - Bitmask for programming languages

**#cpu\_file\_extension** - MCC/MCP cpu file extension string

**#parts** - Bitmask for archiv parts

## 1.31 Languages bitmask

all #languages masks

-----

Bit 0 = English

Bit 1 = German

Bit 2 = French

Bit 3 = Spanish

Bit 4 = Portuguese (European)

Bit 5 = Danish

Bit 6 = Italian

Bit 7 = Dutch

Bit 8 = Norwegian

Bit 9 = Swedish

Bit 10 = Polish

Bit 11 = Finnish

Bit 12 = Hungarian

Bit 13 = Greek

Bit 14 = Esperanto

Bit 15 = Turkish

Bit 16 = Latin

Bit 17 = Russian

Bit 18 = Czech

Bit 19 = Catalanian

Bit 20 = Croatian

Bit 21 = Serbian

Bit 22 = Slovak

Bit 23 = Slovenian

Bit 24 = Portuguese (Brazilian)

---

## 1.32 CPU bitmask

#cpu

----

-1 = unknown

0 = 68000

1 = 68010

2 = 68020

3 = 68030

4 = 68040

5 = 68060

## 1.33 FPU bitmask

#fpu

----

-1 = unknown

0 = NOFPU

1 = 68881

2 = 68882

3 = FPU40

## 1.34 Document type bitmask

#docmask

-----

Bit 0 = ASCII

Bit 1 = AmigaGuide

Bit 2 = HTML

## 1.35 Programming languages bitmask

#plangmask

-----

Bit 0 = C

Bit 1 = C++

Bit 2 = Assembler

Bit 3 = Amiga-E

Bit 4 = Modula-II

---

Bit 5 = Oberon  
Bit 6 = ARexx  
Bit 7 = Cluster  
Bit 8 = Blitz Basic 2  
Bit 9 = MUIBuilder3

### 1.36 MCC/MCP cpu file extension string

#mcc\_cpu\_file\_extension / #mcp\_cpu\_file\_extension

-----

"" | "000"

"010"

"020"

"020f"

"030"

"030f"

"040"

"040f"

"060"

### 1.37 Archiv parts bitmask

#parts

-----

Bit 0 : MCP

Bit 1 : Catalogs

Bit 2 : Docs

Bit 3 : Demo

Bit 4 : Developer

Bit 5 : Source

Bit 6 : Images

### 1.38 Useful text strings

Useful text strings

-----

When using your own text strings, please localize them dependend on #default\_language - otherwise use english texts only.

#prompt\_copy

"Do you want to copy the following file(s)?"

---

#prompt\_makedir

"The following directory was not found, so it must be created:"

#button\_abort

"Abort Uninstall"

#help\_askoptions

"With an Installer prior to version 42.6 it can happen that you see checkbox-buttons without labels. Selecting these buttons will be ignored.\n\n"

#help\_askchoice

"With an Installer prior to version 42.6 it can happen that you see radio-buttons without labels. Selecting these buttons will result in the next lower available selection.\n\n"

#abort\_nocpu

"Sorry, the script is unable to install " @app-name ".mcc, because your CPU configuration is not supported.\n"

#prompt\_uninstall\_parts

"Which parts do you want to uninstall?\n"

@app-name ".mcp will uninstall relating catalogs too.\n"

#prompt\_update

"Which parts of " @app-name ".mcc do you want to add to the installation?"

---