

BumpRevision

Bert Wynants

COLLABORATORS

	<i>TITLE :</i> BumpRevision		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Bert Wynants	November 19, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	BumpRevision	1
1.1	BumpRevision Guide 1.0 Copyright © 1995 B. Wynants	1
1.2	Introducing BumpRevision	1
1.3	golded	2
1.4	projectexample	2
1.5	sourceexample	3
1.6	headerexample	3
1.7	catalogexample	4
1.8	System Requirements	4
1.9	How to use BumpRevision	5
1.10	bump	6
1.11	version	6
1.12	revision	6
1.13	projectname	6
1.14	noasm	7
1.15	noc	7
1.16	existing	7
1.17	beta	8
1.18	file	8
1.19	comment	8
1.20	About the program	9
1.21	About the Author	9

Chapter 1

BumpRevision

1.1 BumpRevision Guide 1.0 Copyright © 1995 B. Wynants

Table of Contents:

Welcome to the BumpRevision{ub} AmigaGuide®. Select any of the following ↔
topics:

Introducing BumpRevision

System requirements

How to use BumpRevision

About the program

About the author

Program History

1.2 Introducing BumpRevision

BumpRevision is a small utility to control versions and revisions ↔
of C and ASM
programs and catalog files. It can be run from the shell only and looks a lot like ↔
all the other revision
bumpers already available.

Why would you use BumpRevision instead of any other revision bumper? Well...
BumpRevision is a very simple yet flexible utility that has all the features you ↔
want.

- o It generates an
include file
(.h or .i) in which it stores the following
information about your program :

- version number
- revision number
- date string
- vers string in several formats
- author name
- project name

If it is missing some information according to your needs. You can add that ↔
information
yourself and BumpRevision will keep your added things in the file.

If it contains too much information, just delete the bits you don't like and ↔
BumpRevision
won't add them again.

- o It has a beta counter.
- o It can keep history information in your
 - sources
 - (asm and c),
 - header
 - files and
 - catalog
 - description files.
- o Included is an ARexx script to be used with GoldED
 - . If you make scripts for other editors please
 - mail them to me for including.

1.3 golded

Install this script in the folder 'GoldED:Arexx'.

add a menu Item named 'Do Revision' and set it to arexx : golded:arexx/DoRevision
add a menu Item named 'Bump Project' and set it to arexx : golded:arexx/DoRevision ↔
BUMP

Selecting 'Do Revision' asks you for a revision comment and adds it to the file ↔
history.

This item won't do anything if the current active window is not a source, header ↔
or catalog file.

Selecting 'Bump Project' will bump the project files in the directory of the ↔
current active window.

1.4 projectexample

```
C example
#define VERSION      40
#define REVISION     4 /* B 1 */
```

```

#define DATE          "12.09.95"
#define VERS          "TIFF 40.04 B 1"
#define VSTRING       "TIFF 40.04 B 1 (12.09.95)\r\n"
#define VERSTAG       "\0$VER: TIFF 40.04 B 1 (12.09.95)\r\n"
#define AUTHOR        "Bert Wynants"
#define PROJECT        "TIFF"
  ASM example
VERSION EQU          40
REVISION EQU         4 ;* B 1
DATE MACRO
  dc.b '12.09.95'
  ENDM
VERS MACRO
  dc.b 'TIFF 40.04 B 1'
  ENDM
VSTRING MACRO
  dc.b 'TIFF 40.04 B 1 (12.09.95)',13,10,0
  ENDM
VERSTAG MACRO
  dc.b 0,'$VER: TIFF 40.04 B 1 (12.09.95)',0
  ENDM
AUTHOR MACRO
  dc.b 'Bert Wynants'
  ENDM
PROJECT MACRO
  dc.b 'TIFF'
  ENDM

```

1.5 sourceexample

```

/** $Revision Header *** Header built automatically - do not edit! *****/
**
** © Copyright Bibbersoft
**
** File           : CLASSBASE.C
** Created on      : dinsdag, 12-sep-95
** Created by     : Bert Wynants
** Current revision : V 1.00
**
** Purpose
** -----
**   TIFF datatype
**
** Date           Author           Comment
** =====
** 12-sep-95     Bert Wynants      --- Initial release ---
**
** $Revision Header *****/

```

1.6 headerexample

```

/** $Revision Header *** Header built automatically - do not edit! *****/
**
** © Copyright Bibbersoft
**
** File           : classbase.h
** Created on     : dinsdag, 12-sep-95
** Created by    : Bert Wynants
** Current revision : V 1.00
**
** Purpose
** -----
**   TIFF datatype
**
** Date          Author           Comment
** =====
** 12-sep-95    Bert Wynants      --- Initial release ---
**
** $Revision Header *****/

```

1.7 catalogexample

```

; ** $Revision Header *** Header built automatically - do not edit! *****/
; **
; ** © Copyright Bibbersoft
; **
; ** File           : deutsch.ct
; ** Created on     : dinsdag, 12-sep-95
; ** Created by    : Bert Wynants
; ** Current revision : V 1.00
; **
; ** Purpose
; ** -----
; **   Datatype Preference Program
; **
; ** Date          Author           Comment
; ** =====
; ** 12-sep-95    Bert Wynants      --- Initial release ---
; **
; ** $Revision Header *****/
## version $VER: DTPPrefs.catalog 1.00 (12.09.95) by Bert Wynants

```

1.8 System Requirements

BumpRevision should run on any Amiga system with at least 512K RAM and one disk-drive. BumpRevision requires KickStart v2.04 or higher to run and is currently a CLI only program.

Should you have any trouble running BumpRevision on your machine, please

write to me
with the full specifications of your machine, that is KickStart

version, model, expansion boards etc...

1.9 How to use BumpRevision

BumpRevision can be started from the Shell only:

Setup the following environment variables:

AUTHOR: SetEnv AUTHOR "your name"

COMPANY: SetEnv COMPANY "your company"

CLI usage:

Usage: BumpRevision BUMP/S, FILE/M/K, COMMENT/K, VERSION/K/N, REV/K/N, PROJECTNAME/ ↔
K,
NOASM/S, NOC/S, BETA/S, RESTART/S

Template:

BUMP/S
increase revision or beta number.

FILE/M/S
Keep history information in file.

COMMENT/K
history log information.

VERSION/K/N
Explicit Version number.

REV/K/N
Explicit revision number.

PROJECTNAME/K
Name of the project.

NOASM/S
Don't generate ASM Include file.

NOC/S
Don't generate C Header file.

BETA/S
Generate beta information.

RESTART/S restarts history or project information.

EXISTING/S
only existing projectfiles are used.

Once everything is set up typing 'BumpRevision BUMP' is enough to increase the ↔
version numbers.

ex:

```
l>BumpRevision PROJECTNAME TestProject BUMP
```

this creates TestProject_rev.h and TestProject_rev.i

bumping the number from now on is done with:

```
l>BumpRevision BUMP
```

1.10 bump

BUMP/S

This argument specifies that the Project files (_rev.h and _rev.i) must be created ↔
or if
they exist that the Revision or beta number must be increased.

1.11 version

VERSION/K/N

This argument specifies the version number (the number before the dot).

You should increase the version number only when major changes and improvements ↔
are done.

It is only needed when you want to specify a new version number.

1.12 revision

REV/K/N

Explicitly set the revision to a specific value. Normally BumpRevision gets the
old revision from the file and increments it by 1.

1.13 projectname

PROJECTNAME/K

This argument should be the name of your project. It will be used
for the naming of the C header file ('projectname_rev.h') and the assembly
include file ('projectname_rev.i').

Is is also used for building the version strings found in the header/include files, so keep care of the right capitalization.

It is needed only the first time, from then on you don't need to specify it. if you don't specify it BumpRevision will scan your current directory and look for an existing project file.

1.14 noasm

NOASM/S

Prevents BumpRevision from generating an include file for 68k assembly language. Useful if version information is only used in C source codes.

Only used when

BUMP
is specified.

1.15 noc

NOC/S

Prevents BumpRevision from generating a header file for C language. Useful if version information is only used in assembly source codes.

Only used when

BUMP
is specified.

1.16 existing

EXISTING/S

Prevents BumpRevision from generating a project file that did not exist. e.g. You only have the _rev.h and you specify EXISTING, BumpRevision will not create the _rev.i file.

Use this as a smart

NOC
and
NOASM
Only used when
BUMP
is specified.

1.17 beta

BETA/S

Enables the beta version count. If BETA is given the first time the revision is incremented as on a normal call. The beta count starts at 1.

On every following call with BETA, only the beta count will be incremented, the revision will stay the same unless specified on the command line.

If the beta stage of the program is finished call BumpRevision without BETA and the beta count will be removed.

HINT

Using the beta switch avoids to increment the revision on each compilation, therefore giving shorter revision numbers for your programs public releases.

1.18 file

FILE/M/K

Keeps an history in your source, header or catalog description files.

This argument can handle multiple files and pattern matching. Thus specifying BumpRevision FILE #? will do all files in the current directory.

This is pretty safe because it won't touch unknown files and will only add the comment to your sources, headers or catalog description files.

If no header is found it will create one. And the supplied comment will be used to set the purpose of the file. The created header can be moved in the file as long as you keep the beginning within the first 10 lines.

If the header has some lines you don't like just throw them out.

If you wan't to add lines you are allowed too.

Never change the following lines:

```
/** $Revision Header *** Header built automatically - do not edit! *****
** Date           Author           Comment
** =====      =====
** $Revision Header *****/
```

NOTE:

When putting a revision header into a catalog description file make sure the ↔
projectname is
supplied (or make sure there is a <project>_rev.h or <project>_rev.i file in the ↔
folder.

1.19 comment

COMMENT/K

Specifies the log info to be added to the History header.

If no header is found it will create one. And the supplied comment will be used to set the purpose of the file.

You should use this if you use the FILE keyword.

1.20 About the program

Current version 1.0

This program is freeware, this means that you can copy it freely as long as you don't ask any more money for it than a nominal fee for copying. If you want to distribute this program you should keep this document with it. This program cannot be used for commercial purposes without written permission from the
author
.

BumpRevision has proven to be stable in everyday use. The author is not responsible for any loss of data, damages to software or hardware that may result directly or indirectly from the use of this program. The author reserves the right to make changes to the software or documentation without notice.

None of the files of the BumpRevision package may be modified. Crunching or achieving is allowed only if none of the BumpRevision files get modified by it.

Special permission is hereby granted to include BumpRevision in Public-Domain collections such as Fred Fish's Amiga Library.

1.21 About the Author

If you have suggestions or remarks about this program, or if you find any bugs, please let me know.

When sending in bug reports, please state exactly under what circumstances the bug occurred, what equipment was used and what happened. If possible also try to give me enough information to reproduce the bug. It is very difficult to find bugs when you don't know exactly what happened.

Write to the following address:

email : bert.wynants@innet.be

Fido : 2:292/603.76 (Bert Wynants)

SnailMail: Wynants Bert
Hogeheide 6
B-2260 Heultje-Westerlo

Belgium - Europe

ENDNODE

Changes & Fixes:

1.0 12-09-95 First release