

SoftConfig

A software package manager

Copyright © 1994 by Denis GOUNELLE.

Any commercial usage or selling without author's written authorization is strictly forbidden. You can copy and spread this program under the following conditions:

1. All the files must be provided
2. No file must be modified
3. You can't charge more than \$6 for the copy

In spite of several tests, no warranty is made that there are no errors in SoftConfig. YOU USE THIS PROGRAM AT YOUR OWN RISK. In no event will I be liable for any damage, direct or indirect, resulting of the use of SoftConfig.

1 Introduction

SoftConfig is a tool both for programmers and users, which aim is to make programs installation and uninstallation easier. It is a complement to the *Installer* program from Commodore, allowing to describe the software configuration of a program, to install some resources (like assigns) at boot time, and to remove a complex software package with a single command.

The basic idea beside SoftConfig is that any program uses some software resources like libraries, fonts, etc... Any installation script must check if these resources are present, and eventually install them. But the more resources a program needs, the more difficult it is to remove this program when you don't need it any more, or when you want to install a new version.

The programmer should never forget that the user might just taking a look at the program, without knowing if he (or she) is going to install it forever. If there is no way to uninstall the program, the user might either throw the whole program away without looking at it, or became very angry when discovering, a few monthes later, that some files are still hiding on his (or she) hard disks.

As a side effect, SoftConfig also simplifies software installation because it will make all assigns and environment variables definition when the Amiga is started. This means that the 'S:User-Startup' file won't be crowded of "assign" commands.

SoftConfig is "pure" and can be made resident.

2 Basics

SoftConfig recognizes the following software resources:

1. directories
2. environment variables
3. assigns (normal and multi-directory)
4. fonts
5. files (this also covers programs, libraries, handlers, ...)

Each resource is defined by it's type, a logical name (an unique identifier), a physical name (the name of the resource on the hard disk), and an eventual version number. For example, we can define the `'reqtools.library'` by:

```
type : file
logical name : reqtools.library
physical name : LIBS:reqtools.library
version number : 38
```

The logical name is very important, because it will be used for any further reference to this resource. In order to avoid confusions, I suggest that you use the "basename" (i.e. the physical name of the file or directory without it's path).

SoftConfig maintains a database of all declared resources. The name of the database is `'S:SoftConfig'`. It is strongly advised to make regular backup of this file, and to copy it in a safe place before starting any installation using SoftConfig. This way, if something goes wrong, you will be sure that the resource base is not corrupted.

3 Commands

SoftConfig LIST [type] [FULL]

Displays all the resources in the database. The optional "type" argument can be used to display only the resource of the corresponding type, and may take one of the following value :

- DIR (directories)
- VAR (env. variables)
- ASSIGN (normal assign)
- ASSIGNADD (multi-dirs assign)
- FONT (font)
- FILE (others)

The optional "FULL" argument can be used to display more informations about each resource.

SoftConfig INSTALL

This command should be added to your 'S:User-Startup' file, in order to perform all startup initializations. This way, SoftConfig will automatically "install" assigns and variables resources.

SoftConfig REMOVE <logical name> [VERIFY]

Uninstalls the given resource. This also removes any resource that is not used by another resource. **Caution ! Any physical file corresponding to the removed resources will be deleted. For directory and assign kind resources, this means that all the files and sub-directories in the corresponding directory will be deleted too.**

If the optional "VERIFY" argument is specified, you will be asked to confirm each deletion.

SoftConfig FIND <logical name> <type> [version]

Searches the resource database for a resource which has the given logical name and type. Returns 0 (OK) if the resource is found, 5 (WARN) if not. An optional version number may be specified, and will be considered as the minimal version number of the resource to find.

SoftConfig ADD <logical name> <physical name> <type> [version]

Adds a resource in the database. Depending on the resource's type, the physical name is expected to be:

FONT the font directory's name (e.g. "FONTS:Topaz")

VAR	the value of the variable
DIR	the directory's name
FILE	the file's name
ASSIGN	
ASSIGNADD	the directory to which the assign points to

SoftConfig ATTACH <logical name> <logical name>

Attaches a sub-resource to a resource. For example, if a program uses a library, you have to add the resource corresponding to the program, then the resource corresponding to the library, and attach the library resource to the program resource.

The first logical name is the master resource name, and the second is the sub-resource name.

SoftConfig DETACH <logical name> <logical name>

Detaches a sub-resource from a resource. This may be useful if a new version of a program doesn't need a resource which was required for the previous version.

The first logical name is the master resource name, and the second is the sub-resource name.

SoftConfig UPDATE <logical name> <new version>

Changes the version number of a resource. The new version number must be greater than the current one.

SoftConfig MOVE <logical name> <new physical name>

Modifies the physical name of a resource. This may be useful if you move a resource to another directory or, for variable and assign-kind resources, if you want to change the "value" of a resource.

SoftConfig DEF <var> <value>

Defines an internal variable named "var", and sets it to the given value. You can then use the variable in another command, with the "{var}" form. Note that all variables are lost when SoftConfig terminates, so variables are mostly usefull within an script file.

SoftConfig FROM <input file>

Reads a set of commands from the given file (see Chapter 4 [Example], Page 5 for a sample script).

4 Example

This example shows the SoftConfig actions required by my other program ABackup.

The program is made of an executable file, with an icon file. It uses the 'xpk.library', and the 'ABCK:' assign. Let's suppose that the user decided to install ABackup in the 'SYS:Utilities' directory, and to assign 'ABCK:' to the 'S:ABackup' directory.

Here is the corresponding (commented) script for SoftConfig :

```
;
; Setup variables
;
DEF ProgDir SYS:Utilities/
DEF LibsDir LIBS:
DEF ABCKDir S:ABackup
;
; Declare ABackup as the "main" resource
;
ADD ABackup ProgDirABackup FILE 4.03
;
; Declare the icon file as a sub-resource
;
ADD ABackup.info ProgDirABackup.info FILE
ATTACH ABackup ABackup.info
;
; Declare the xpk.library as a sub-resource
; The library is made of a ".library" file and of a sub-directory,
; so we first attach the library to ABackup, then the sub-directory
; to the library.
;
ADD xpk.library LibsDirxpkmaster.library FILE 2.4
ATTACH ABackup xpk.library
ADD xpk-compressors LibsDirCompressors DIR
ATTACH xpk.library xpk-compressors
;
; Declare the ABCK: assign, as a sub-resource
;
ADD ABCK: ABCKDir ASSIGN
ATTACH ABackup ABCK:
;
```

You should execute this script, then try some commands like "LIST" or "REMOVE". Changing the variables values will allow you to make your tests without destroying too much data...

5 Final notes

I hope this tool will be considered as usefull by all amiga users, and programmers will start to include it as part of their installation procedure.

Criticisms and suggestions will always be welcomed. Write to:

M. GOUNELLE Denis
27, rue Jules GUESDE
45400 FLEURY-LES-AUBRAIS
FRANCE

You can also send a message to the following Internet address : "gounelle@alphanet.ch". Note that this mailbox is not mine, so please send only short messages. As I don't have direct access to the messages, don't expect an answer before a dozen of days.

SoftConfig is written in C language, and was developped on an Amiga 3000 UNIX-1 (10 Mb RAM, internal HD disk drive, two internal hard disks of 100 Mb and 160Mb, external SyQuest drive of 88 Mb, external floppy disk drive, 1960 monitor and Star LC24-10 printer) connected by a null-modem cable to an A500 with 1 Mb RAM.

Table of contents

1 Introduction 1

2 Basics 2

3 Commands 3

4 Example 5

5 Final notes 6