# PSXControl

Tim Jacobs and Korneel Ketelslegers Schächtelin

| COLLABORATORS |
|---|

| | | | |
|---|---|---|---|
| | *TITLE* :<br>PSXControl | | |
| *ACTION* | *NAME* | *DATE* | *SIGNATURE* |
| WRITTEN BY | Tim Jacobs and Korneel Ketelslegers Schächtelin | August 7, 2022 | |

| REVISION HISTORY |
|---|

| NUMBER | DATE | DESCRIPTION | NAME |
|---|---|---|---|
| | | | |

# Contents

# Chapter 1

# PSXControl

## 1.1 AmigaGuide Developer documentation to PSXControl v1.0

```
                            - -·    Faeries Geneses    ·- -

                 - --·        brings you         ·-- -


                - - --$\div${·   PSXControl v1.0   ·}$\div$-- - -

                       - -$\div$ Developer Docs $\div$- -


                 - -$\div$         choose one:        $\div$- -
```

Important to know is that this doc is not complete. Loads of stuff will change in future releases! At the moment, only the library calls are

```
    fixed, so don’t rely on any other values.
                Contact Tim Jacobs
                 for more info

 - -$\div$ "The biggest trick the devil ever pulled was convincing everybody" $\ ↩
    div$- -
 - -$\div$ "                  that he doesn’t exist.                        " $\ ↩
    div$- -

                - -$\div$ Verbal Kint (The Usual Suspects) $\div$- -
```

## 1.2 Disclaimer

```
                  - - --$\div${·   PSXControl v1.0   ·}$\div$-- - -

                       ·        DISCLAIMER         ·
```

1. Tim Jacobs, Korneel Ketelslegers and Tobias Schächtelin, further referred to as ’the authors’, cannot be held responsible for anything that might go wrong when using this package.

2. This  package is guaranteed to be 100% (one-hundred percent) virus-free upon release. If there is a virus in this package then it was placed there by someone else!

3. The authors of this package cannot be blamed if this particular piece of software does not function correctly on your system.

4. We hereby reject any liability or responsibility for these or any other consequences from the use of this software whatsoever. This includes, but is not limited to, damage to your equipment, to your data, to other machines your amiga is connected to, any equipment connected to those hosts, personal injuries, financial loss or any other kinds of side effects.

5. This software is provided as-is. This means we do not guarantee that this software is fit for any specific purpose and we do not guarantee any bug fixes, updates or help during error recovery.

6. All product names, company names are copyrights and trademarks of their respective owners.
   * Playstation and PSX are trademarks of Sony Computer Entertainment Inc.
   * Net Yaroze is a trademark of Sony Computer Entertainment Inc.
   * SIOCONS is a trademark of Sony Computer Entertainment Inc.
   * Macintosh is a trademark of Apple Inc.

## 1.3 Introduction

```
                  - - --$\div${·   PSXControl v1.0   ·}$\div$-- - -
```

· INTRODUCTION ·

PSX-Control also offers the possibility for other coders to take advantage of the Playstation's capabilities. To help those coders (and not telling them to re-invent the wheel) some basic functions are included in PSX-Control that make their lifes a bit easier.

Some of those functions include transfering and execution of files.

Imagine you want to code a 3D-object viewer (with texturemapping and phong-shading and all those goodies), but you simply can't get it fast enough on the Amiga. Then the solution is simple: You transfer a homemade viewer executable to the Playstation, you put all the texture and object data on one PSX memory address and then run the program! If the user wants to view other objects, then you simply transfer new data to that address and re-execute the program.

This is just an example of thousands of possible projects you can think of.

Want to code a render plugin for your favourite renderprogram so you can see your rendered scenes immediatelly and realtime ? Or want to walk through your virtual dreamhouse or want to see how your house looks like when a plane crashes into it ? All this is now possible with the render capabilities of the Playstation.

The main advantage of PSX-Control is that the util coder doesn't have to write his/her own software for every different type of Playstation development system there is. All these can be adressed in the same way using the psxamiga.library

## 1.4 Requirements

- - --$\div${·   PSXControl v1.0   ·}$\div$-- - -

· REQUIREMENTS ·

In order to develop for PSX-Control you at least need an Amiga that is capable of running PSX-Control :).

Knowledge of how to use libraries can also come in handy, but if you want to develop some serious applications, knowledge of Playstation programming in general could be usefull.

## 1.5 Using the psxamiga.library

- - --$\div${·   PSXControl v1.0   ·}$\div$-- - -

· USING THE LIBRARY ·

Using the psxamiga.library is very simple: You open the library and call the functions you need! There are some things you REALLY should pay

```
   attention to though.

   Things to check in the psxamiga library base

   * Before calling a function, always check the pa_Status. This must
     be set to UBYTE 1 before you call a function. If it is set to UBYTE 0,
     then another program is already performing an action that locks the
     other programs out (eg. transfering of files). If another program
     is using the psxamiga.library, then the function calls will simply
     fail.

   The library functions

   * Transfering of files is done by calling the function Transfer2PSX().
     This function automatically detects what kind of file you are transfering
     and will act accordingly.

     Executables will always be transfered to the CORRECT address. If you
     want to upload code at random addresses, then DON'T include the PSX-EXE
     or ECOFF header in that file, and specify the correct destination address
     (remember that PSX Executables are always hardcoded for a certain
     address. On the PSX the rule:
     "coders that write from a fixed address should be shot"
     is not valid.

     Data files are transfered to the address specified.

  * If you want to execute a file, then first set all registers to the correct
    addresses using the function SetCPUReg() and then call the function
    PSXExecute(). The registers you should AT least set are:

      EPC = The program counter. The Playstation will jump to this address when
            you do a PSXExecute().
      GP  = Global Pointer (usually set to zero)
      SR  = Stack Register (usually set to $801ffff0)
```

## 1.6 DevInfo

```
                  - - --$\div${·   PSXControl v1.0   ·}$\div$-- - -

                           ·  PLAYSTATION DEVELOPMENT  ·

   Some important things you should know about Playstation development are:
     * Never transfer a file to the adressing range $80000000 - $0801FFFFF
       when the pa_System is YAROZE. The Yaroze terminal and LIBPS.EXE files
       reside at these addresses and overwriting them will make the Yaroze
       crash.

     * Take care when transfering files to certain addresses. If several
       programs are using the psxamiga.library at the same time, then those
       programs might overwrite addresses that you are using.
```

## 1.7 The Authors

- - --$\div${·   PSXControl v1.0   ·}$\div$-- - -

·   THE AUTHORS!   ·

The PSXControl program and all the accompanying files where created
by the Faeries Geneses team.

To contact us for bugreports, suggestions, please send an email to:

faeries@gmx.net

You can also contact Faeries Geneses members if you like. Currently our
team consists of the following members:

THE TEAM

- Tim Jacobs
  Main coder and public relations manager.

  E-Mail : crisp@unicall.be

  Currently studying physics at the Limburgs Universitair Centrum (LUC)
  in Diepenbeek as a first-year student.

- Korneel Ketelslegers
  Coder and visual art director.

  E-Mail : malcomn@gmx.net

  Currently studying film-making techniques at the art department of
  the KHLim (Katholieke Hogeschool Limburg) in Genk as a fourth-year
  student.

- Tobias Schächtelin
  Main coder and Supervising Producer

  E-Mail : grey@gmx.net

  Currently studying literature and linguistics at the University of
  Tübingen.

OTHER PROJECTS

This is a list of programs that have been released by Faeries Geneses:

* MP3Tagger v1.3
  Simple and effective editing of information within MP3 audio files.
  Released on aminet as "mus/misc/FG-MP3Tagger.lha".

Other projects currently in development:

* conversion.library (finished)
  Easier converting of numbers from ascii <-> hex, dec, oct, bin
  for asm and c programmers.

* Mogaira (abandoned)
  Converts autodocs to amigaguide files, .i includes to .h includes, fd files
  to pragma & protos files.

* ww_Addnote (80% finished)
  Adds footnotes to WordWorth documents faster and more confortable.