

ShellID

By Jason Gouger

Copyright © 1995 All rights reserved

COLLABORATORS

	<i>TITLE :</i> ShellID		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	By Jason Gouger	October 23, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ShellID	1
1.1	ShellID By, Jason Gouger	1
1.2	Disclaimer	2
1.3	Introduction	2
1.4	ShellID's Many Features	3
1.5	Installation	3
1.6	Local Environmental Variables	3
1.7	Launching ShellID	4
1.8	Designing Scripts	5
1.9	Example Scripts	6
1.10	Known Problems	7
1.11	ShareWare	7
1.12	Comments and Suggestions	8
1.13	Other Doors by Jason Gouger	8
1.14	History	9
1.15	Registration * To print registration form select Print from the Project Menu *	9

Chapter 1

ShellID

1.1 ShellID By, Jason Gouger

```
*****
*                                     *
*           ShellID Version 1.00      *
*                                     *
*       Written By, Jason Gouger[M3012] *
*   © Copyright 1995 All rights reserved *
*                                     *
*****
```

```
+-----+
| THIS DOOR IS SHAREWARE, see
|   ShareWare
|   notice |
+-----+
```

```
=====
= Note: Since registrations have been VERY slim on my other doors, =
= I have added an annoying feature to ShellID. In order to =
= enter ShellID one must answer a question to encourage the =
= sysop to register the door, the response MUST be 'Yes'. =
= Furthermore, the response must be COMPLETELY typed, a 'Y' =
= will NOT be accepted. =
=====
```

~~~~~Disclaimer~~~~~

~~~~~Introduction~~~~~

~~~~~Features~~~~~

~~~~~Installation~~~~~

~~~~~Known~Problems~~~~~

~ShareWare~Registration~

Comments~and~Bug~Reports

~~~~Other~Doors~By~Me~~~

~~~~~History~~~~~

~~~Registration~Form~~~

1.2 Disclaimer

Disclaimer:

~~~~~

In the paragraphs below 'author' refers to Jason Gouger and 'program' refers to ShellID (c) Copyright 1995, by Jason Gouger, All Rights Reserved.

This program may only be reproduced and used in accordance with the requirements and limitations set forth below. This program may be used by licensees of MEBBSNet Professional BBS software in combination with their lawful use of licensed copies of MEBBSNet Professional BBS software. This software may be redistributed on a non profit basis by any official MEBBSNet CSR support site. Any other distribution sites MUST the express written permission to do so by the author. Redistribution of this program for profit requires the express, written permission of the author. Furthermore, all advertising materials mentioning the features, or use, of this software must acknowledge the author and software. This archive MUST be distributed in an UNALTERED form.

The Author makes no warranties or representations, either express, or implied, with respect to the information set forth herein, which the information is being supplied on an "AS IS" basis, subject to change without notice. The ENTIRE risk as to the use of this program is assumed by the user. IN NO EVENT WILL THE AUTHOR OF THIS PROGRAM BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY CLAIM ARISING OUT OF THE INFORMATION PRESENTED HEREIN, EVEN IF IT HAS BEEN ADVISED OF THE POSSIBILITIES OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE LIMITATION OF IMPLIED WARRANTIES OR DAMAGES, SO THE ABOVE LIMITATIONS MAY NOT APPLY.

'MEBBSNet Profession BBS' is a registered trademark of MEBBS Engineering and Software.

## 1.3 Introduction

Introduction:

~~~~~

If you have been waiting for a interactive shell handler for MEBBSNet, your wait is over. ShellID is a STDIO interpreter for MEBBSNet. It is a complete STDIO handler which interfaces directly to MEBBSNet. Therefore it does NOT depend on other external handlers such as the FIFO: device. Several local environmental variables are set to facilitate the operation of Shell Based Doors.

1.4 ShellID's Many Features

Features

~~~~~

- independent program with internal handler
- creates several local environmental variables for use with shell based games
- ability to have an interactive shell
- allows the use of control codes
- smart carrier drop routines to terminate the task.

## 1.5 Installation

Installation:

~~~~~

Files included in the archive:

- 1) ShellID/ShellID =====> Shell Interpreter (Door File)
- 2) ShellID/ShellID.SCP =====> Example Remote Shell Script
- 3) ShellID/ShellID.guide =====> ShellID Documentation.

Place the above files in a Directory Called ShellID in the Doors Path configured in MEBBSNet.

Note: ShellID uses the Doors Path as DEFINED IN MEBBSNet, you MUST have this configured correctly. This documentation will assume that you have it configured for 'DOORS:'. If you have it configured otherwise just remember 'DOORS:' refers to the path defined IN MEBBSNet.

*** ShellID REQUIRES AmigaOS 2.04+ ***

More information:

Local~Enviornental~Variables

Launching~ShellID~~~~~

Designing~Scripts~~~~~

Example~Scripts~~~~~

1.6 Local Enviornental Variables

Local Enviornental Variables:

~~~~~

1. UserName.... Set to user's REAL Name
2. Handle..... Set to user's Handle (if configured in MEBBSNet)
3. MinsLeft.... Set to user's remaining time

4. LineNumber.. Set to the line number on which the user located

Other information can be set with the InfoID program. Contact your nearest MEBBSNet CSR to receive a copy of InfoID.

More information:

Launching~ShellID~~~~~

Designing~Scripts~~~~~

Example~Scripts~~~~~

## 1.7 Launching ShellID

Launching ShellID

~~~~~

When ShellID launches a NewShell, a script named [DoorName].SCP will be executed. For example if your menu entry calls DOORS:ShellID/ExampleDoor, the script named DOORS:ShellID/ExampleDoor.SCP will be executed. Notice that the PATH and DoorName is completely used, with the .SCP extension appended.

MEBBSNet should call the DoorFile ShellID, which can be either copied to other door names in order to launch different scripts, or I recommend using the AmigaOS MakeLink command instead of copying the door file. I will show you exactly what needs to be typed in order to use the MakeLink command, if you have any further questions on the MakeLink command consult your AmigaOS manual. The MakeLink command basically makes a new directory entry pointing to another file on the disk. Therefore it does not require additional disk space.

Example:

To make a link to the ShellID program to execute a Script Called MyGame.SCP, you would type in the DOORS:ShellID Directory:

```
1> MakeLink MyGame ShellID
```

This would make a link from ShellID to the a door called MyGame. With this setup it is ONLY necessary to update the ACTUAL ShellID door file when installing newer versions of ShellID. ALL linked file names will use the NEW version of ShellID.

In the Previous Example you would have the following menu entry:

Option '*' :

```
Function MultiRundoor DOORS:ShellID/MyGame
end
```

This would launch the MyGame Door File (Linked to ShellID) and which in turn would execute the script called DOORS:ShellID/MyGame.SCP in the NewShell. With this setup you can have an unlimited number of scripts without using additional

disk space for each DoorFile.

More information:

Designing~Scripts~~~~~

Example~Scripts~~~~~

1.8 Designing Scripts

Designing Scripts

~~~~~

The general guidelines for designing scripts can be found in your AmigaOS manual. I will go over the security requirements to ensure a secure shell.

- 1) Make SURE you have a High FailAt Value (2147483647 Max Value).  
Basically you want to ensure that programs launched within the shell will not halt the script due to a return code.
- 2) Setup an infinite EndCLI loop.

Example:

---> Start --->

```
FailAT 2147483647
```

```
;
; Place your desired command(s) here
;
```

```
Lab ExitScript
  EndCLI
Skip ExitScript Back
```

<---- End <----

Notice the FIRST line of the script sets the FailAT condition and the end part of the script sets up an infinite EndCLI loop to ensure that the EndCLI is executed.

More information:

Example~Scripts~~~~~

## 1.9 Example Scripts

Example Scripts:  
~~~~~

Note: The local environmental variables UserName, Handle, MinsLeft, and LineNumber are preset with the online user's information.

Example Script For Dave's Space Empire (DSE):

```
---> Start --->

FailAt 20000                ; Setting Failat Condition
                            ;
DOORS:DSE                   ; Changing Directory to DOORS:DSE
DSE -s "$Handle"           ; Starting DSE with User's Handle
                            ;
Lab ExitScript              ; Setting up infinite loop label
  EndCli >NIL:              ; Calling EndCLI, to exit Shell
Skip ExitScript back        ; Looping to try again if EndCLI Failed

<---- End <----
```

Example Script For Hack 'N Slash (H&S):

```
---> Start --->

FailAt 20000                ; Setting Failat Condition
                            ;
DOORS:H&S                   ; Changing Directory to DOORS:H&S
HSClient -t $MinsLeft "$Handle" ; Starting H&S w/ MinsLeft and Handle Info
                            ;
Lab ExitScript              ; Setting up infinite loop label
  EndCli >NIL:              ; Calling EndCLI, to exit Shell
Skip ExitScript back        ; Looping to try again if EndCLI Failed

<---- End <----
```

Example Script For Al's BackJack Casino (XBJ):

```
---> Start --->

FailAt 20000                ; Setting Failat Condition
                            ;
DOORS:XBJ                   ; Changing Directory to DOORS:XBJ
XBJ -u "$Handle"           ; Starting XBJ with Handle Info
                            ;
Lab ExitScript              ; Setting up infinite loop label
  EndCli >NIL:              ; Calling EndCLI, to exit Shell
Skip ExitScript back        ; Looping to try again if EndCLI Failed

<---- End <----
```

Example Remote Shell Script:

---> Start --->

```
Execute S:Shell-Startup          ; Assuming you have a default Shell Startup.
                                ; Notice there is NO security measures since
                                ; we are establishing an interactive shell.
```

<---- End <----

1.10 Known Problems

Known Problems:

~~~~~

Due to the limitations of the MEBBSNet IO Door Handler, some doors may not function properly. The problem will occur when a shell program attempts to use the STDIO port asynchronously. However most doors do NOT use this method, as it is more difficult method to integrate into programs. Currently ShellID is NOT able to provide an asynchronous environment for programs. However, ShellID \*IS\* designed to handle this mode, but with the limitations of MEBBSNet it is not reliable to integrate this mode. If this mode would be integrated into the operation of ShellID, MANY characters would be 'lost' due to the fact that MEBBSNet will NOT hold characters, and the door MUST be WAITING for the character to arrive. Thus, if the character appears while ShellID is processing IO from the Task, the character from MEBBSNet would be lost. For this reason ShellID has been linked to MEBBSNet in a synchronus manner.

As soon as MEBBSNet supports IO in a 'proper' manner I will integrate the changes to the door. When such changes are made, this door WILL be a 100% working shell interface. However, until that time ShellID should work with over 95% of the shell based doors.

Note: Several CTRL Codes including the Escape key can NOT be retrieved from the user. This again is a limitation of the functions I am using to reliably retrieve information from MEBBSNet.

## 1.11 ShareWare

ShareWare:

~~~~~

This door is SHAREWARE, so if you use it for more than 15 days, please register your copy. You will receive a key file which will remove the registration question from the program startup. Also, you will receive a 1-Year subscription status on my BBS which will entitle you to Unlimited Downloads. Make sure you send a password if you would like an account setup.

Updates can be Freq'ed from 1:268/114.0 with the magical name 'ShellID' and a password of 'UPDATE'.

More information:

Comments~and~Suggestions

Show~Registration~form~~

1.12 Comments and Suggestions

Comments and Bug Reports:

~~~~~

Call my BBS:

The Amiga Enterprise      717-223-8269      300-28800 Bps      24 Hours

Or send mail to:

gouger@psu.edu            - Internet Address \*

gouger@cse.psu.edu - Internet Address (Talk Cmd Supported) \*

Jason Gouger            - Fidonet 1:268/114

Jason Gouger            - Proper Mebbs Echo

\* Internet addresses are the fastest method of communication

## 1.13 Other Doors by Jason Gouger

Other Doors by Me:

~~~~~

MQwkID [V1.00]

MQwkID is an external door for the MEBBSNet Profession BBS software that creates standard QWKMail packets. MQwkID has many features including autopacket naming, multiple message area downloads, hidden mode operation, and much more. MQwkID also serves as a reply packet tosser directly through the door or from the command line.

UserID [V2.01]

UserID keeps track of the past number of callers that YOU define. Also the output is 100% configurable. There are many text sequences (~u_) which display a callers actions while on-line. They include messages posted, download, and upload information. Every field can be configured to a specific length and can be left or right justified. UserID will create an environmental variable called User_X (X being the line number) which will hold the on-line user's name. This environmental variable can be used for shell games and other applications where the username is not readily available, see the included examples. The environmental variable is also used by the TimeOn door. The TimeOn door displays the total on-line time.

CDRomID [V1.02]

Allows users to search CD-Rom Index file(s) for a string, and of course it has the ability to the allow direct downloading of files from the CD-Rom. It also includes the ability to make file(s) FREE for download, or have them charged against the users account. It now has Multiple CD-Rom support, and a temporary download directory option has been added.

LineID [V1.00]

LineID displays a configurable number of user's online. The output is 100% configurable. This configurability is accomplished through several text sequences. If LineID is used in conjunction with UserID a SIGNIFICANT increase in speed will be noticed. I recommend installing this wonderful last caller door.

PathID [V1.00]

PathID is a small utility to ensure all of the file, message, and system directories have been created. If a missing directory is encountered PathID will prompt to create the missing directory if desired. This is an excellent utility for first time setups, and when creating new file or message areas.

InfoID [V1.00]

InfoID is a small utility to generate local variables containing most of the information on a particular user. The variable names are 100% configurable, and a line number or username may be specified to generate the user information. This utility may be used to initialize CLI doors with a user's personal preferences, or scripts that process user information.

1.14 History

History:

~~~~~

===> 1.00 ===>

First Public Release.

## 1.15 Registration \* To print registration form select Print from the Project Menu \*

ShellID Registration

~~~~~

Please provide the following information, and include a Money order or check for \$15 (US Funds ONLY) and send it to:

Jason Gouger
74 White Birch Lane
East Stroudsburg, PA 18301-8339

I can only accept personal checks in US dollars drawn on US banks. Money orders are preferred and will result in faster processing.

=====

Name : _____

Street : _____ Apt : _____

City : _____ State/Prov: _____

Country: _____ Zip/Postal: _____

Email : _____

BBS Name: _____ BBS number: (____)____-____

Sysop Name: _____ (MUST MATCH MEBBSNet Config)

Send Keyfile: () UUEncoded Internet Mail

- () UUEncoded FidoNet Mail
() Deliver NetMail to node: 1:_____ (US Only)
() Hold keyfile at node: 1:268/114.0 (Outside US)
for node: _____

- () Pick it up on my BBS (717-223-8269)
BBS Account Information: Phone number: _____
Password: _____

- () Postal Mail Service. (Add \$5.00 for S/H and Disk)

Comment: (Write on back of form if needed)

Date : _____ Signature: _____