Tim Whitson
The Fool's Quarter – fqbbs.synchro.net

August 11, 2021

GameSrv Setup with Synchronet BBS Software under Windows 10 32bit

GameSrv is an awesome program written by Rick Parrish of R&M Software. It can be run as a standalone door game server or as a server for your BBS. I always loved door games the most about BBSing in the 90's. It was the main reason I initially set one up myself, and now run one today.

To quote Rick, "This is why GameSrv was born, to give SysOps a simple yet powerful way to host door games for generations to come!"

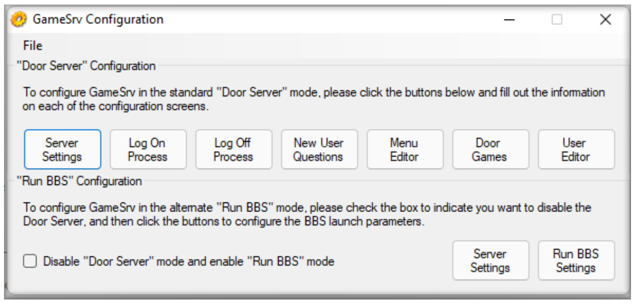
I run my BBS, The Fool's Quarter, under Windows 10, 32 Bit, using Synchronet BBS software by Rob Swindell. Synchronet has great door game support and is extremely easy to set doors up on but for some reason I started playing with GameSrv and liked it. I did find I can get a few of the older games running on GameSrv that I could not on Synchronet. This could just be me…

Here is how I set up GameSrv. I'm sure there are other ways to set it up, but this works for me.

Download GameSrv from: https://www.gamesrv.ca

Unzip it into its own directory. I put mine on the same computer where my BBS is running, directly on the C drive. Nice thing about modern computers is they are way overpowered for running a BBS. It can easily run the BBS and the game server.

RUN: GameSrvConfig.exe and select "Server Settings" button to configure your server.



"Door Server" Configuration

To configure GameSrv in the standard "Door Server" mode, please click the buttons below and fill out the information on each of the configuration screens.

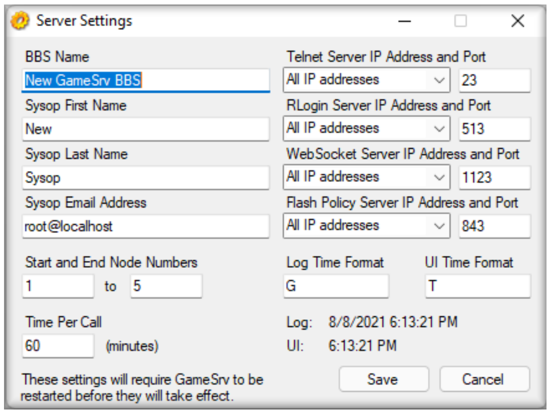
"Run BBS" Configuration

To configure GameSrv in the alternate "Run BBS" mode, please check the box to indicate you want to disable the Door Server, and then click the buttons to configure the BBS launch parameters.

☐ Disable "Door Server" mode and enable "Run BBS" mode

If you are running GameSrv on the same computer as your BBS it is very important you change the default telnet port from 23 to something else. Your BBS and GameSrv cannot be on the same port. I set mine to 2323, and do not PORT FORWARD this port if you don't want people to log directly into your game server. I also changed the Rlogin port to 512, Synchronet is set at 513.

On the drop-down menu for the IP Address it should show the IP address for the computer you are installing it on, select this address.


Server Settings

BBS Name	New GameSrv BBS	Telnet Server IP Address and Port	All IP addresses, 23
Sysop First Name	New	RLogin Server IP Address and Port	All IP addresses, 513
Sysop Last Name	Sysop	WebSocket Server IP Address and Port	All IP addresses, 1123
Sysop Email Address	root@localhost	Flash Policy Server IP Address and Port	All IP addresses, 843
Start and End Node Numbers	1 to 5	Log Time Format / UI Time Format	G / T
Time Per Call	60 (minutes)	Log: / UI:	8/8/2021 6:13:21 PM / 6:13:21 PM

These settings will require GameSrv to be restarted before they will take effect.

The majority of the options in the configuration gives you a message: Still need to implement this. But you can edit these areas manually, I'll explain later.

Once you answer all the questions and save the information it will generate a filed called: "gamesrv.ini" in the config directory. You can either edit this file manually or use GameSrvConfig.exe for future changes.

Logon Process:

You want to edit the file: logonprocess.ini found in the config directory. This tells GameSrv what to do when people log on. Even though the docs for GameSrv are lacking, the notes included in the individual *.ini files will make up for them.

This one is pretty easy to understand. First it will display a welcome ansi file, with a pause to continue. If you don't want to use this you can either delete it or comment it out. I recommend commenting the lines out using semi-colons. You never know when you might want something here.

The next command sends the user to the MAIN MENU, unless they have less than standard security level. I'm not going to get into that.

The logonprocess.ini sends you to the main menu. If you notice there is a Parameters line that says: Parameters=MAIN. That will tell the program to go to the MENUS directory and use the main.ini file.

Main Menu:

Go to the MENU directory you will find a file called main.ini. This is where you can really start customizing your game server. GameSrv will create canned screens or you can develop your own main.ans file to display. If there is a file called main.ans in the MENUS directory it will be displayed when the logonprocess.ini calls the Main Menu. One thing to note, the main.ans is independent of the main.ini. For example, you can have a "T" option on the main.ans to run the game Tradewars. If it is not in the main.ini to have "T" run Tradewars it will not run. On the other side of that, if you do not have a "T" on the main.ans file to run Tradewars, but it is set up in the main.ini, if someone presses "T" it will run Tradewars. The ansi file is just like pretty wrapping paper.

Since it is just being used as a game server, my Main Menu is where I sort out the games by genre, fantasy, space, cards and such. It has commands to jump back to the BBS, but not to log off the BBS. This requires caller to transfer back to the BBS to log off completely.

Valid options for the Action= line are:

<u>ChangeMenu</u>
 This will move the user to another menu
 The Parameters= line should be the name of the new menu (and an .ini with that name should exist in the menus subdirectory)
<u>Disconnect</u>
 This will immediately disconnect the user
 The Parameters= line does nothing for this action
<u>DisplayFile</u>
 This will display a file to the user
 The Parameters= line should be the path and filename of the file (can be absolute, or relative to GameSrv directory)
<u>DisplayFileMore</u>
 Same as DisplayFile, but will show a MORE prompt every 24 lines
<u>DisplayFilePause</u>
 Same as DisplayFile, but will wait for a key to be pressed after the file is displayed
<u>LogOff</u>
 This will initiate the log-off process
 The Parameters= line does nothing for this action
<u>Pause</u>
 Pause execution for a variable number of seconds
 The Parameters= line defines the number of milliseconds to pause for (1000 milliseconds equals 1 second)
<u>RunDoor</u>
 Execute a door program
 The Parameters= line defines which door to run. A matching .ini file must exist in the doors subdirectory
<u>Telnet</u>
 Telnets to a remote server
 The Parameters= line should be the hostname:port (default 23 if port not specified) of the remote server to connect to

To transfer to the Fantasy Game Menu, I use the following in the main.ini file:

```
[F]
Name=Fantasy Menu
Action=ChangeMenu
Parameters=FANTASY
RequiredAccess=10
```

[F]	- Key on Main Menu to change to Fantasy Menu
Name	- The name that GameSrv will display on generic menus
Action	- Action Code to change to a different menu
Parameters	- The parameter is what menu name to change to
RequiredAccess	- Security access required to use this function

This will open the FANTASY Menu using the fanstasy.ini and displaying fantasy.ans if it exists.

As an example, I will run through installing the game Death Masters on my BBS under the Fantasy Menu. I will expand on each of these steps below.

In the MENU directory:

1. Add a key to the fantasy.ini file.
2. Modify fantasy.ans screen to show callers the game is available.

In the DOORS directory:

1. Create a death.ini file.
2. Make a directory called "death" and unzip the game files.
3. Set up the game.

MENU Directory:

Add the following to the fantasy.ini:

```
[D]
Name=Death Masters
Action=RunDoor
Parameters=DEATH
RequiredAccess=10
```

[D]	- Key on Main Menu to change to Fantasy Menu
Name	- The name that GameSrv will display on generic menus
Action	- Action Code to run a game in the DOORS directory
Parameters	- This will run the death.ini file in the DOORS directory
RequiredAccess	- Security Level required to access this menu option.

Modify your fantasy.ans file. I like Moebius as a screen editor. There are several SyncDraw and the old time favorite TheDraw.

DOORS Directory:

When the “D” key is pressed on the Fantasy Menu it runs the death.ini file. These files are the the DOORS directory. The *.ini file for the game has to be called the same as the parameter you pass. Parameters=DEATH then your *.ini file has to be called: death.ini. See the _sample.ini file on page 7 for other parameters and such.

death.ini:

```
[DOOR]
Name=Death Masters
Command=DOORS\DEATH\START.BAT
Parameters=*NODE
LocalOutput=True
Native=False
```

[DOOR]	- This cannot be changed from [DOOR]
Name	- The name that GameSrv will display on generic menus
Command	- Changes directory and runs the batch file
Parameters	- You can pass various parameters to the game
LocalOutput	- This lets you snoop on the people playing games
Native	- Please read on this, DOOR32.SYS games would be TRUE

Create a directory called “death” and unzip the game files into. I am going to assume you understand all the complexities of setting up door games. They are like mouse traps, everyone seemed to have a better idea.

I set up all my games to run from a batch file called START.BAT. It makes things easier when setting up, especially if you have lots of door games.

Here is a copy of one of my START.BAT files, this one is for Death Masters:

```
@ECHO OFF
if %1 == 1 goto NODE1
if %1 == 2 goto NODE2
if %1 == 3 goto NODE3
if %1 == 4 goto NODE4
goto END

:NODE1
   CD DOORS\DEATH
   DEATH.EXE C:\GAMESRV\NODE1\DOOR.SYS
   GOTO END

:NODE2
   CD DOORS\DEATH
   DEATH.EXE C:\GAMESRV\NODE2\DOOR.SYS
   GOTO END

:NODE3
   CD DOORS\DEATH
   DEATH.EXE C:\GAMESRV\NODE3\DOOR.SYS
   GOTO END

:NODE4
   CD DOORS\DEATH
   DEATH.EXE C:\GAMESRV\NODE4\DOOR.SYS
  GOTO END

:END
   EXIT
```

I am not a batch file wizard. I know you can write them differently. A couple of lines is really all you need with variables. My 32bit games use the variables so are a lot shorter. However, this way helps me set up the nodes differently as needed, especially when it comes to single node games and having to lock the game.

If you open the file _sample.ini in the DOORS directory, it will give you a instructions on how to set up the door game *.ini files, parameters you can pass and such.

_sample.ini

```
 [DOOR]
Name=Legend Of the Red Dragon
Command=doors\lord\start.bat
Parameters=*NODE
Native=False
ForceQuitDelay=5
WatchDTR=True
WindowStyle=Minimized
```

The **[DOOR]** line must appear exactly as is (don't call it [LORD] or something like that)
The **Name**= line is what will be displayed to the user when using the 'canned' menus.
  It will also be displayed in the admin interface
The **Command**= line is what will be executed to run the door.
  This should be the path (absolute, or relative to the GameSrv directory) and filename of the command to execute.
  Don't include command-line parameters here, they come next
The **Parameters**= line contains the optional parameters to pass to the door.
  This can be hardcoded values, or command-line specifiers (see below for a full list)
The **Native**= line controls how the door is executed.
  Old 16bit DOS doors should set this to False, modern 32bit doors should set this to True
  NB: If you want to use NetFoss, you need to set this to True, since NetFoss is a 32bit program (NetFoss is no longer included, so you'll have to set that up on your own)
The **ForceQuitDelay**= line controls how long (in seconds) GameSrv will wait for the door to terminate after a hangup event
  If the user hangs up while in a door, GameSrv will detect that and wait up to this many seconds for the door to gracefully terminate
  If the door doesn't terminate after this many seconds, it will be force-quit to ensure the node isn't locked up indefinitely
The **WatchDTR**= line controls whether GameSrv will watch for (and honour) DTR drops by the door
  If you want GameSrv to disconnect the user if the door drops DTR, then set this to true
  If you want GameSrv to keep the user connected if the door drops DTR, then set this to false
  The default is true, but if for example you want to run a BBS from within GameSrv, then you'll probably want to set this to
  false so the user returns to GameSrv after the BBS quits.
  NOTE: This setting only applies when running 16bit doors on 32bit Windows
The **WindowStyle**= line controls how the output of the door is displayed.
  Valid options (which should be self explanatory) are:
    Hidden, Minimized, Normal, Maximized

Using an invalid option will cause the external to fail to run. CASE IS IMPORTANT so use exactly as typed above

I do not use the last three lines, ForceQuitDelay, WatchDTR, and WindowStyle, as you can see by my example above.

The Command-line specifiers that are passed on the Parameters line are:

*ALIAS - ***The user's alias
*DOOR32 - Full path\filename to the door32.sys drop file
*DOORSYS - As above, for door.sys
*DOORFILE - As above, for doorfile.sr
*DORINFO - As above, for dorinfo.def (no node number in filename)
*DORINFO1 - As above, for dorinfo1.def (filename always contains 1, no matter which node)
*DORINFOx - As above, for dorinfox.def (where x is the current node number)
*HANDLE - The connection's socket handle (deprecated; may be removed in future versions)
*IPADDRESS - The remote user's ip address
*MINUTESLEFT - The number of minutes the user has remaining (ie 5 minutes 10 seconds will report as "5")
*NODE - The current node
*PASSWORD - ***The user's password (by default this will be the hash, unless you disabled hashing in gamesrv.ini in which case it'll be the actual password)
*SECONDSLEFT - The number of seconds the user has remaining (ie 5 minutes 10 seconds will report as "310")
*SOCKETHANDLE - Same as *HANDLE
*USERNAME - Same as *ALIAS
Anything in the newuser.ini file - ***For example if you have an [Email] entry in newuser.ini, you can use *EMAIL to pass that on the command-line.

I'm not going to reproduce the entire security note in the _sample.ini file but please read it. I don't think it's needed if you are only using GameSrv as a game server to Rlogin from a BBS.

Normally I only pass the *NODE parameter. For the native 32 bit games I have, like Ambroshia, I also pass *SOCKETHANDLE.

If you pass *NODE as the first parameter it will be %1, the next parameter will be %2 and so on.

I've not tried to pass any dropfile information. GameSrv will make a directory called NODE*, where * is the node number people are calling into when someone logs on. In this directory it will write various generic door drop files that can be passed to the game. Once the person logs out of GameSrv the directory is deleted. You can call the drop file from this directory, or if needed copy the dropfile to the game directory as some games require.

As I stated earlier most of the features in the GameSrvConfig do not work. If you need to edit a user file you can go to the USERS directory and see a list of the user *.ini files. This file can be changed as needed, increase or decrease the security level for example:

<u>User File in *.ini format</u>:

```
[USER]
AccessLevel=100
Alias=Tim Whitson
AllowMultipleConnections=False
PasswordHash=XXXX
PasswordSalt=XXXX
UserId=2
```

I deleted the passwords for security reasons.

Custom Menus

You can also create different menus for each access level, so for example if you have an item on the main menu that only sysops with an access level of 100 can see, then you can create a **menus\main.ans** that everyone will see, but also create a **menus\main100.ans** that only users with access level 100 will see. Similarly, if you have a reduced menu that twit users should see, you could create a third **menus\menu0.ans**.

The ansis/bulletins/menus all support using MCI codes, which allow you to embed certain variables into the display. The currently supported codes are:

ACCESSLEVEL	The current user's numeric access level
ALIAS	The current user's alias
BBSNAME	Your BBS name, as defined in the config/gamesrv.ini
DATE	The system's current date, in short format
GSDIR	The directory GameSrv is running from
MENUNAME	The menu the user last loaded
NODE	The node the user signed on to
OPERATINGSYSTEM	The system's operating system
SYSOPEMAIL	Your email address, as defined in config/gamesrv.ini
SYSOPNAME	Your name, as defined in config/gamesrv.ini
TIME	The system's current time, in short format
TIMELEFT	The current users's time left for this call, in hh:mm:ss format

To use the codes, wrap them in { and } characters. So for example, {ALIAS} will display the user's alias.

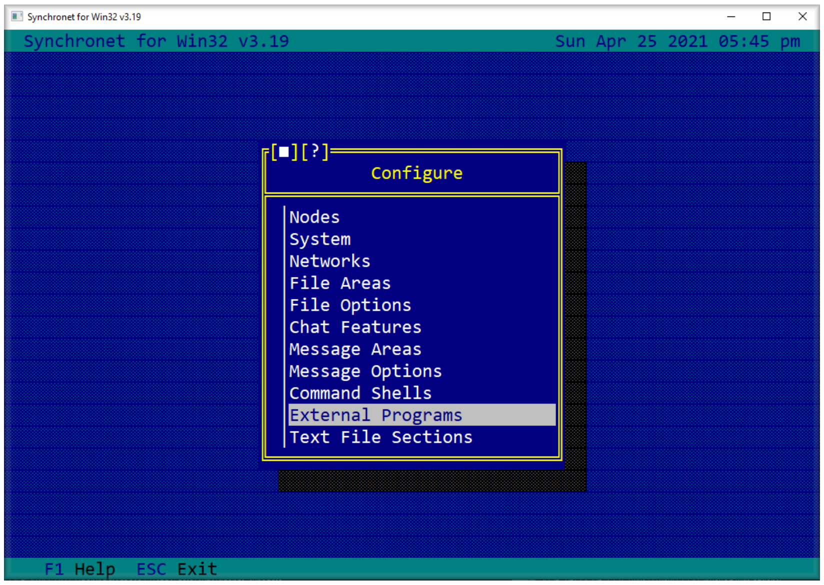
Each code also supports justification of short values. So for example, if you want to display the user's access level and always have it take up 3 spaces on screen, use {ACCESSLEVEL3} to left-align the text and pad the right side with spaces, or {3ACCESSLEVEL} to right-align it and pad the left with spaces.

When values are too long for the requested size, they'll be truncated. So for example, if you want to display the user's alias (which is 'rumpelstiltskin') and always have it take up 10 spaces on screen, use {ALIAS10} to truncate the end of the string leaving 'rumpelstil', or {10ALIAS} to truncate the beginning of the string leaving 'lstiltskin'. (Not that you'd probably ever want to truncate the alias so short, but it's just an example of how it works.)

I think the only one of these I would probably use would be TIMELEFT. I run customized menus and I’d just type in my name or BBS name, not use a code.

Finally, if you want a parting ansi screen or something, this can be called in the logoffprocess.ini. Read through that and adjust as needed. I left my stock.

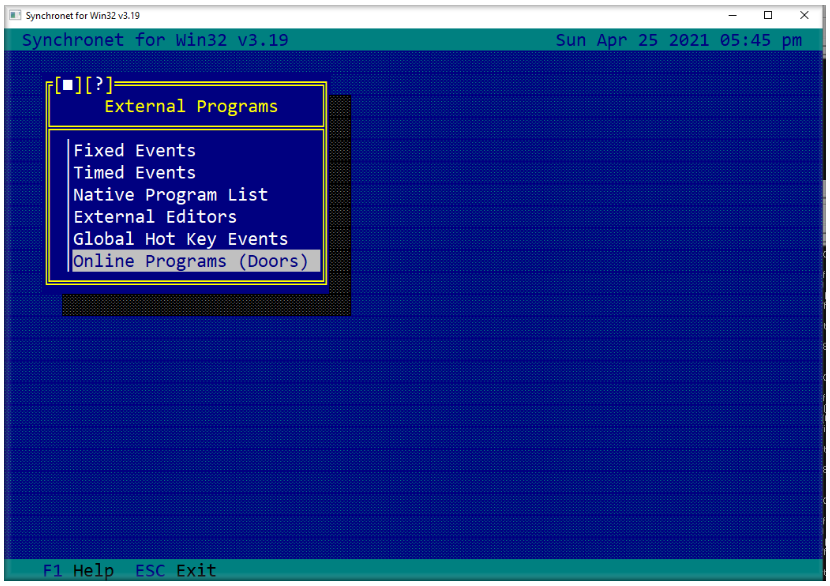
Setting up Synchronet to Rlogin to GameSrv:



```
Synchronet for Win32 v3.19                       Sun Apr 25 2021 05:45 pm

                    [■][?]
                          Configure

                     Nodes
                     System
                     Networks
                     File Areas
                     File Options
                     Chat Features
                     Message Areas
                     Message Options
                     Command Shells
                     External Programs
                     Text File Sections

  F1 Help  ESC Exit
```

<u>Configure Menu</u>

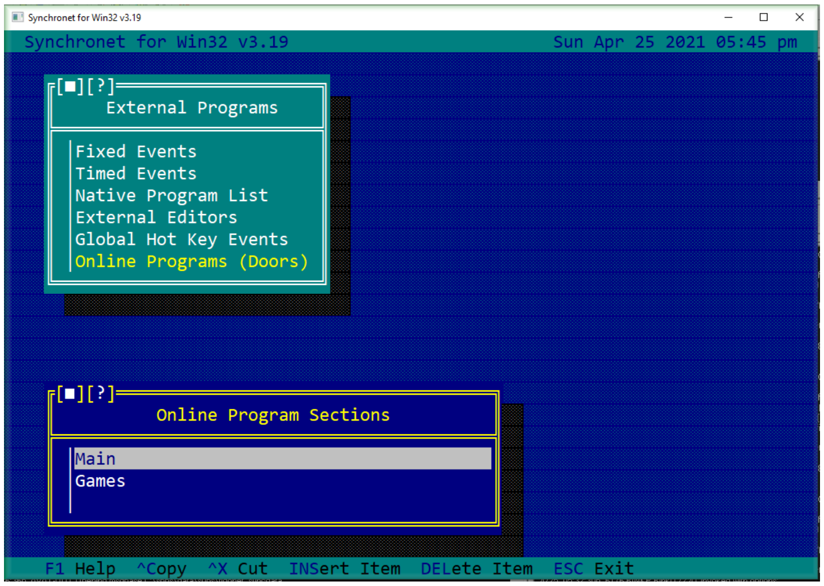
Setting up Synchronet to use RLogin to another server is quite simple. On your Synchronet Control panel select the BBS tab and select CONFIGURE to open the menu. Scroll down and click on EXTERNAL PROGRAMS.



```
Synchronet for Win32 v3.19                       Sun Apr 25 2021 05:45 pm

   [■][?]
        External Programs

    Fixed Events
    Timed Events
    Native Program List
    External Editors
    Global Hot Key Events
    Online Programs (Doors)

  F1 Help  ESC Exit
```

<u>External Programs Menu</u>

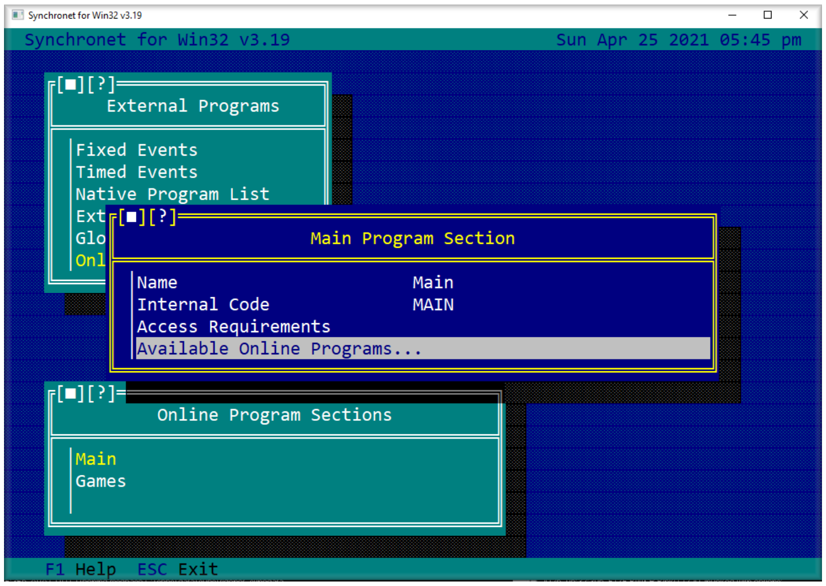
Scroll down and click on ONLINE PRGRAMS (DOORS)



```
Synchronet for Win32 v3.19
Synchronet for Win32 v3.19                              Sun Apr 25 2021 05:45 pm

[■][?]
      External Programs

  Fixed Events
  Timed Events
  Native Program List
  External Editors
  Global Hot Key Events
  Online Programs (Doors)

[■][?]
           Online Program Sections

  Main
  Games

  F1 Help  ^Copy  ^X Cut  INSert Item  DELete Item  ESC Exit
```

Online Programs Menu

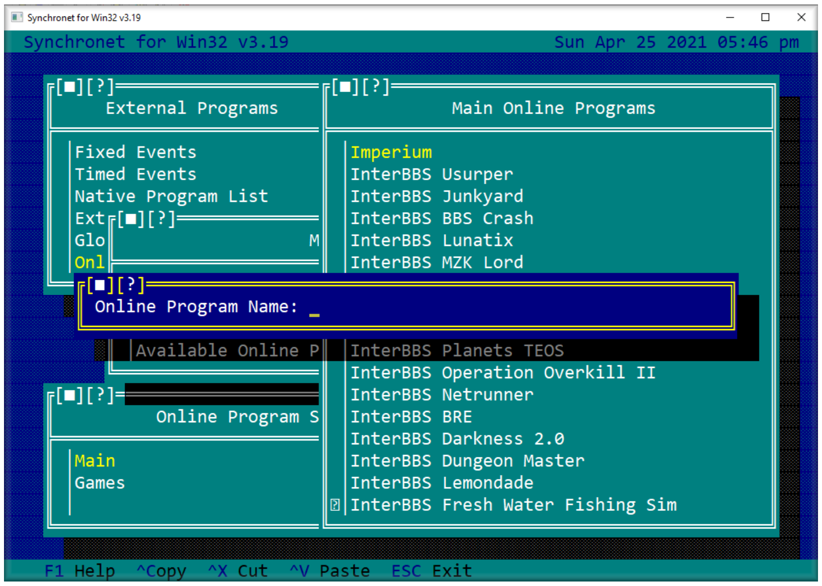
I assigned a key on my Main Menu to access the game server. You can put it on any menu, but for the purposes of this instruction, it is on the Main. Click on MAIN.



```
Synchronet for Win32 v3.19
Synchronet for Win32 v3.19                              Sun Apr 25 2021 05:45 pm

[■][?]
      External Programs

  Fixed Events
  Timed Events
  Native Program List
  Ext[■][?]
  Glo                Main Program Section
  Onl
       Name                             Main
       Internal Code                    MAIN
       Access Requirements
       Available Online Programs...

[■][?]
           Online Program Sections

  Main
  Games

  F1 Help  ESC Exit
```

Main Program Menu

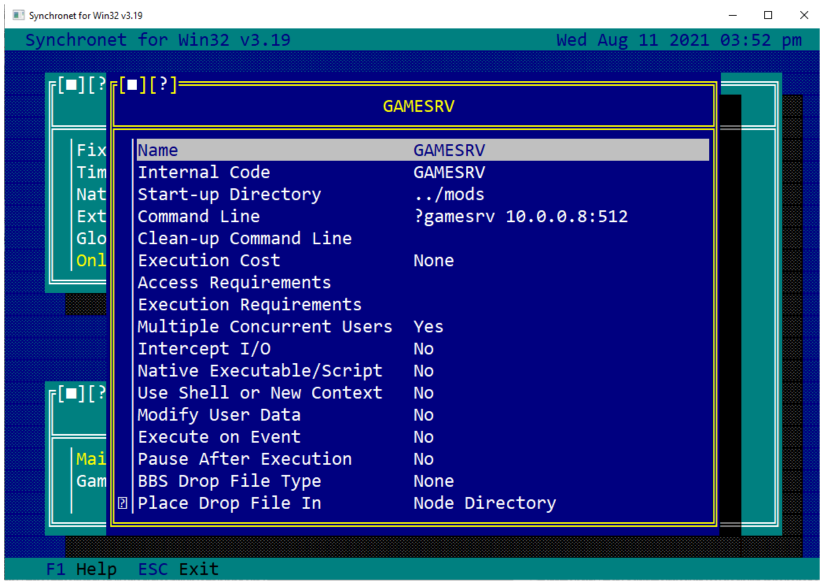
Select the AVAILABLE ONLINE PROGRAMS



```
Synchronet for Win32 v3.19                              Sun Apr 25 2021 05:46 pm

 External Programs                    Main Online Programs
 Fixed Events                         Imperium
 Timed Events                         InterBBS Usurper
 Native Program List                  InterBBS Junkyard
 Ext                                  InterBBS BBS Crash
 Glo                                M InterBBS Lunatix
 Onl                                  InterBBS MZK Lord
   Online Program Name: _
   Available Online P                 InterBBS Planets TEOS
                                      InterBBS Operation Overkill II
                                      InterBBS Netrunner
       Online Program S               InterBBS BRE
                                      InterBBS Darkness 2.0
 Main                                 InterBBS Dungeon Master
 Games                                InterBBS Lemondade
                                      InterBBS Fresh Water Fishing Sim

 F1 Help  ^Copy  ^X Cut  ^V Paste  ESC Exit
```

Main Program Menu

A menu will open to allow you to name the new Online Program.
I named mine: GAMESRV



```
Synchronet for Win32 v3.19                              Wed Aug 11 2021 03:52 pm

                         GAMESRV
 Name                          GAMESRV
 Internal Code                 GAMESRV
 Start-up Directory            ../mods
 Command Line                  ?gamesrv 10.0.0.8:512
 Clean-up Command Line
 Execution Cost                None
 Access Requirements
 Execution Requirements
 Multiple Concurrent Users     Yes
 Intercept I/O                 No
 Native Executable/Script      No
 Use Shell or New Context      No
 Modify User Data              No
 Execute on Event              No
 Pause After Execution         No
 BBS Drop File Type            None
 Place Drop File In            Node Directory

 F1 Help  ESC Exit
```

GAMESRV

The INTERNAL CODE should be autofilled.
In the COMMAND LINE enter: ?gamesrv 10.0.0.2:153 (Enter your computer IP Address)
I set the access level to 10 so a guess account can't log into the server. Set it to whatever your lowest user access you want.

I modified the rlogin.js file slightly and put it in the /SBBS/MODS directory.

```
------------

// gamesrv.js

// Telnet Gateway using RLogin protocol - Requires v3.00c

// $Id: gamesrv.js,v 1.4 2017/10/25 08:59:15 rswindell Exp $

// @format.tab-size 4, @format.use-tabs true

load("sbbsdefs.js");

var flags = 0;
if (argc > 1)
    flags = eval(argv[1]);
bbs.rlogin_gate(argv[0], flags);
console.clear();

------------
```

I wanted it to jump from my BBS to GameSrv seamlessly with no pausing. So I deleted a couple of lines out of the standard rlogin.js that write to screen.

You should always put your modified files in the /SBBS/MODS directory.

Good luck and enjoy!