Technical Analysis of The Sims Online: Pre-Alpha

Jeremy Glazebrook

Nio2so: Server Emulation

Jeremy Glazebrook

Contents

Preface

Terminology used in this document

Terminology used in this document may be related to nio2so directly such as the following:

- **TSOHTTP(S) –** This is the term used for the HTTP(S) servers that served functionality related to logging in and basic avatar data.
    - HTTP was used very sparingly in TSO: Pre-Alpha, basically only being used for Authentication and a basic skeleton of Avatars the User owns.
    - TSOHTTP(S) does direct the client to a web address to connect to when accessing **TSOTCP.**
- **TSOTCP** – This is the term used for the TCP servers used for the City Server.
    - It uses Aries a means of transport and Voltron as the structure of the packets themselves.
- **Voltron –** This is the term used for the underlying protocol/architecture for communication with the City Server and Game Client.
    - **See:** PDU
- **Aries** – Aries is an additional system used by The Sims Online to facilitate the transfer of packets to/from the game client and City Server.
    - **Aries Packet(s)** – Packets formatted for use with Aries Servers and Clients.
- **PDU** – A term used in reference to a Voltron Packet that is handled by a Regulator in The Sims Online.
    - The term means "Protocol Data Unit"
- **Regulators** – A term used to describe a Controller, effectively. It has a Protocol that it uses to interpret a Voltron Packet sent from the City Server / Room Server.
    - *There are numerous regulators in The Sims Online: Pre-Alpha.*
    - These Regulators can also send messages to and from each other running locally in the game client, when it sends outbound communications over the internet is referred to as "Transmitting" a message.
- **kMSGs** – These are the messages that are sent to and from Regulators and can have structures attached to them which are referred to by a CLSID
- **Protocol** – A term used in reference to the available set of packets (PDUs) the Game Client can send/receive to the City/Room server it is connected to.
    - *They have formatting, fields, and expected values that the game client will validate when communicating with the City Server.*
- **RefPack** – A LZ77/LZSS compression algorithm used in games that reference Gimex.
    - See also: RefPack - Niotso Wiki
- **TSOSerializedStream (Stream)** – This is a structure used to send large amounts of serialized binary data in a packet (or many split packets)
    - See also: Stream - Niotso Wiki

TSOHTTP & TSOHTTPS

Emulating the HTTP servers used to facilitate aspects of the Online experience.

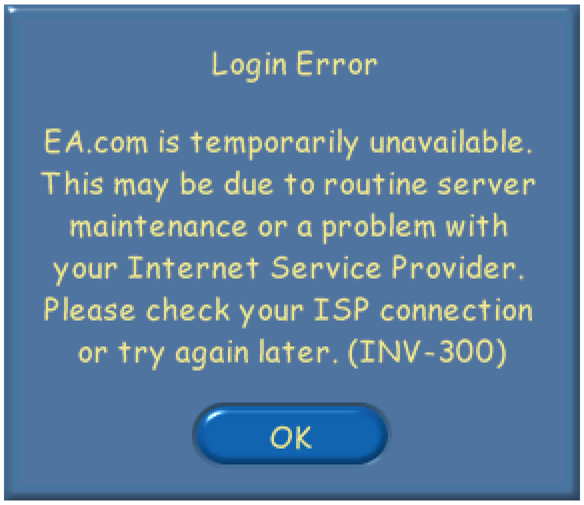
Logging into The Sims Online - AuthLogin

Logging in is handled by the AuthLogin resource. This is located at www.ea.com/AuthLogin for the Pre-Alpha version of The Sims Online.

First and foremost, the game will Ping this resource on using HTTPS. If the Ping is unsuccessful, an error (INV-300) message is displayed *(Figure 0.A)*

If the password does not match or an invalid username is provided, and Ping to the resource is successful, the following error message is displayed. *(Figure 0.B)* Note that the server can dictate what the error message says.

There are other error codes that can be specified as well, which will produce the result shown in Figure 0.B.



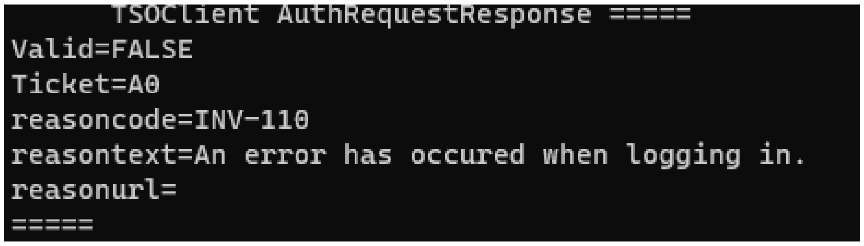
Login Error

EA.com is temporarily unavailable. This may be due to routine server maintenance or a problem with your Internet Service Provider. Please check your ISP connection or try again later. (INV-300)

Figure 0.A

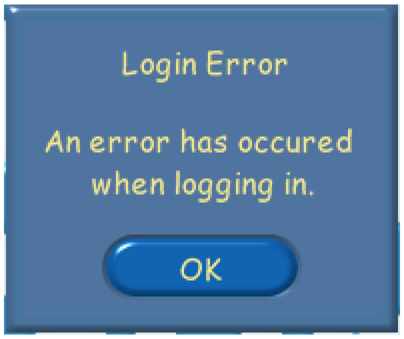
Using nio2so, you can see the result of the HTTPS requests to log in by utilizing the Console attached to the TSOHTTPS process.

The HTTP packet body is XML in nature, and has the following components listed in Figure 0.C.



```
      TSOClient AuthRequestResponse =====
Valid=FALSE
Ticket=A0
reasoncode=INV-110
reasontext=An error has occured when logging in.
reasonurl=
=====
```

Figure 0.C



Login Error

An error has occured when logging in.

Figure 0.B

The HTTP result is always 200 (Ok). If it is not Ok, an error is displayed not matching what the packet dictates. The components are summarized:

1. VALID – Since 200 (Ok) is always used, this acts as the HTTP status code.
2. Ticket – The current ticket awarded to the client. More on this later.
3. Reason Code – This is the error code the game client recognizes. Certain error codes provided will show a hard-coded error message.
4. Reason Text – The error message to display in the event the Game Client does not have that error message hardcoded.
5. Reason URL – Doesn't appear to do anything in this version.

From here, all connection back and forth is handled over an HTTP connection.

Successful Log In Attempts – Initial Connect Servlet

Successfully logging in will allow the client to proceed to the InitialConnectServlet. The Initial Connection Servlet is at the resource: http://xo.max.ad.ea.com/city-selector/initial-connect.jsp

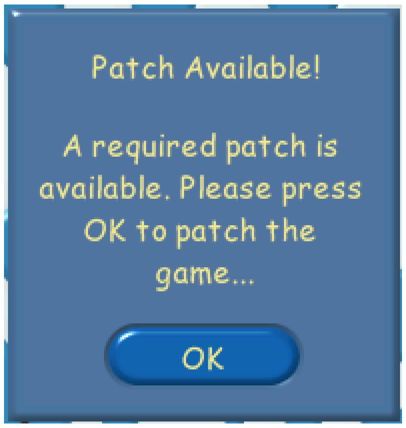
This has three known responses in TSO: Pre-Alpha. Which are detailed below.

Patch Result

A Patch Result is when the game needs to update itself before it can be allowed to connect to a City Server (Shard). The Patch Result is shown in Figure 0.D.

```
     CitySelector: InitConnect(0) ===
<Patch-Result>
       <Authorization-Ticket>Loginnio2soDefault</Authorization-Ticket>
       <Patch-Address>https://nio2so.tso.ad.max.ea.com/patch1.0</Patch-Address>
     </Patch-Result>
```

Figure 0.D


Patch Available!

A required patch is available. Please press OK to patch the game...

OK

Figure 0.E

Patch Result has an XML element of Patch-Result. It tells the Game Client where to connect to, which nio2so specifies as https://nio2so.tso.ad.max.ea.com/patch1.0

The game will prompt the user in Figure 0.E.

Pressing OK will invoke buddy.exe to patch the game.

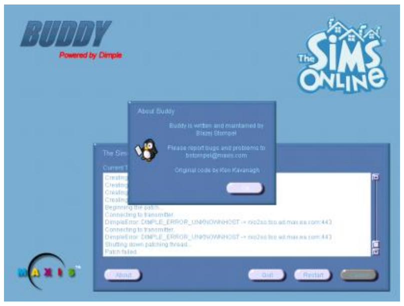

BUDDY
Powered by Dimple
The Sims Online
About Buddy
Buddy is written and maintained by Blazej Stompel
Please report bugs and problems to bstompel@maxis.com
Original code by Ken Kavanagh
About
Quit
Restart

Figure 0.F

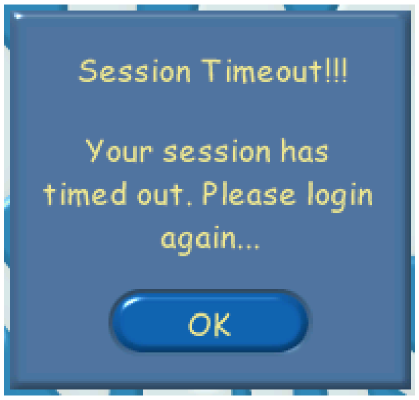
Error Result

An error result is when there is an error in handling the request on the InitialConnectServlet. It is unknown when this could have happened except due to an internal server error.

The error message will be interpreted as: Session Timeout!!!

```
     CitySelector: InitConnect(2) ===
<Error-Message>
       <Error-Number>132</Error-Number>
       <Error>Generic nio2so error.</Error>
     </Error-Message>
```

Figure 0.G


Session Timeout!!!

Your session has timed out. Please login again...

OK

Success Result

If all is successful in the InitialConnectServlet when the Game Client connects to the resource, then a UserAuthorize packet is sent to the Game Client to move on to the next step in logging in to Select-A-Sim.

The UserAuthorize packet has a body with a root element called <User-Authorized/>

Select-A-Sim – The Avatar Data Servlet

After we traverse the InitialConnectServlet the next stop is to visit the AvatarDataServlet. This servlet is responsible for serving the listings of Avatars that are currently on the user's account.

This resource is at http://xo.max.ad.ea.com/cityselector/avatar-data.jsp

Avatar Data has an XML body with the root element being `<The-Sims-Online>`

Inside the root element is the following elements:

1. Avatar-Data
    a. AvatarID – The ID of the Avatar
    b. Name – The Name of the Avatar
    c. Simoleans – How much wealth they have
    d. Simolean-Delta – How much wealth has changed (unknown timestep)
    e. Popularity – How many friends this Avatar has?
    f. Popularity-Delta – Same as above.
    g. Shard-Name – The City Server this Avatar lives on. This is the actual name of the City.
        i. For example: Blazing Falls
2. Player-Active – Optional, no impact.

This will cause an avatar to appear. Sending an empty root element will allow the user to Create an Avatar.

Select-A-Sim – The Shard Status Servlet

The final stop on our journey before entering into Select-A-Sim completely is to serve the resource located at: https://xo.max.ad.ea.com/cityselector/shard-status.jsp

This will tell the game client about what servers are online.

It has the root element <Shard-Status-List>

1. Shard-Status
    a. Location – public/private – Dictates whether it appears in the list of Shards.
    b. Name – The name of the City.
        i. E.g. Blazing Falls
    c. Status – up – The status of the shard.
        i. "up" is the only known good value.
    d. Online-Avatars – The current amount of players online in this Shard.

You can add as many servers as you like to this list.

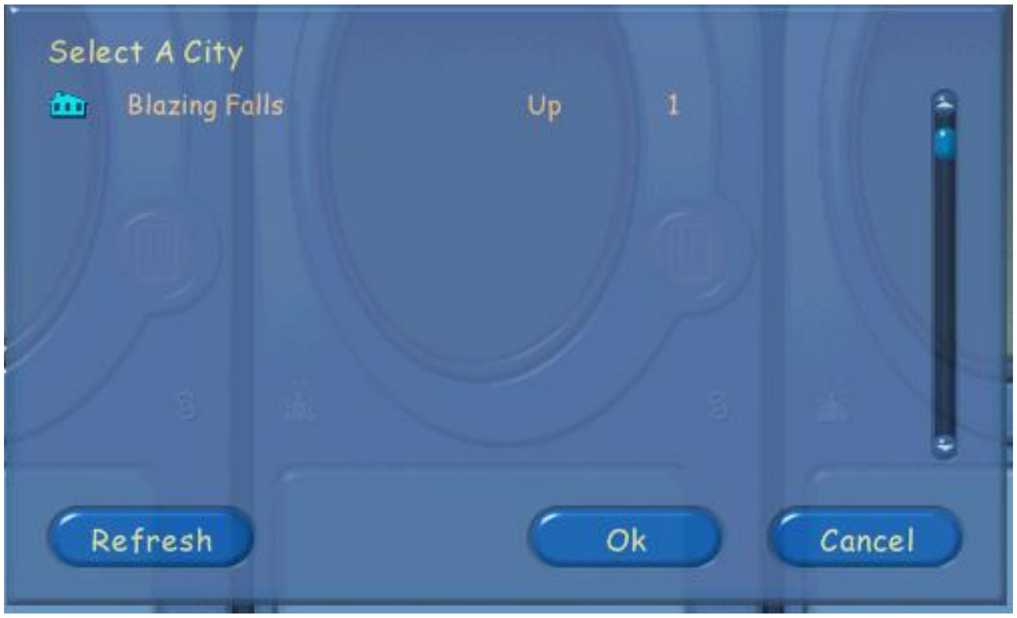
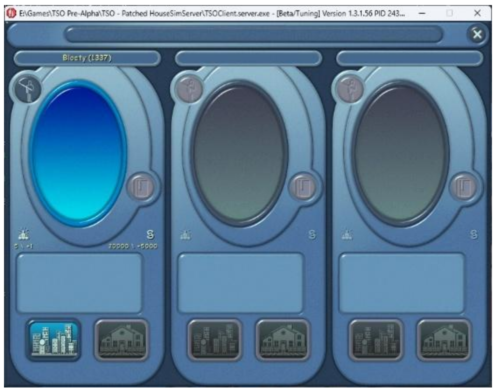


Figure 0.H – Shards listed with 1 player online and the name, Blazing Falls.


E:\Games\TSO Pre-Alpha\TSO - Patched HouseSimServer\TSOClient.server.exe - [Beta/Tuning] Version 1.3.1.56 PID 243...
Bloaty (1337)

Select-A-Sim – Selecting a Sim and Creating a New One

From here, (almost) all connectivity is handled through TSOTCP (nio2so terminology) using the City Server.

In this version of TSO, the player is only able to create one Avatar. The remaining two slots are available if you manually add Avatars to the account, but the button will only appear on the first slot.

The City Server – Initial Connectivity

Emulating the City Server which was used to create the gameplay experience.

Clicking "Create an Avatar" on the first slot or selecting an existing Avatar will connect to the City server located on the Avatar's Data.

TSOHTTP – The Shard Selection Servlet

TSOHTTP makes its final request here, invoking the Shard-Selection Servlet.

The Shard-Selection Servlet will tell the game client the address to connect to, to access the City Server. It has the root element <Shard-Selection> and the following elements:

1. Connection-Address – The address to connect to.
    a. The port is required here. The port will ALWAYS have 100 appended to it automatically.
    b. *localhost:36 will be interpreted as: localhost:36000*
2. Authorization-Ticket – This is that ticket awarded to client in the AuthLogin section.
3. PlayerID – The ID of the Avatar selected.

If the PlayerID is 0, this indicates the Client is connecting to the City Server to access Create-A-Sim

Aries Client Session Info Packet

Before we start communicating with the Game Client once it makes a successful connection to TSOTCP, we first ask it for its client version and user info.

To do this, we use Aries Packets. Aries Packets are binary in nature, they do not have XML formatting like TSOHTTP did. See more about Aries Packets here.

Once a response to the Client Session Info Query packet has been received, we can move onto the next packet.

Connecting to the City Server

The series of communication in packets is still under active research.

Here is what is known so far:

Nio2so Method:

Sender	Voltron Packet Type	Description
Server	NA	Aries Client Session Info Query Packet
Client	NA	Aries Client Session Info Packet
Server	0x1D	Host Online PDU
Client	0x09	Client Online PDU
Server	0x3C	Update Player PDU
Client	0x33	Set Ignore List PDU
Client	0x37	Set Invisible PDU
Client	0x35	Set Invincible PDU
Client	0x31	Set Accept Alerts PDU
Client	0x43	Set Accept Flashes PDU
Client	0x92	BC Version List PDU
Client	0x99	Load House Response PDU
Server	0x97	House Sim Constraints Response PDU
Client	0x82 … 0x5BB73FAB	Database Request PDU – Get Char Blob By ID Request
Server	0x82 … 0x5BB73FE4	DB – Get Char Blob By ID Response PDU
Client	0x82 … 0xFD8F9080	Database Request PDU – Get Bookmarks Request
Server	0x82 … 0x3D8F9003	DB – Get Bookmarks Response PDU

Niotso “Mashuga” Method:

The method used for the Mashuga client written for niotso:

Sender	Voltron Packet Type	Description
Server	NA	Aries Client Session Info Query Packet
Client	NA	Aries Client Session Info Packet
Server	0x1D	Host Online PDU
Server	0x3C	Update Player PDU
Client	0x09	Client Online PDU

The Sims Online World View (MapView)

Updated 7/2/2025

Nio2so has made amazing progress in the MapView. This section outlines what can be accomplished in the MapView.

Data Types

The various types of data found in The Sims Online: Pre-Alpha.

Please observe that this information is due to the work of niotso and is here for preservation purposes in the event the original resource is made unavailable.

Aries Packets

Aries Header

These are packets formatted with an Aries Header. The header consists of the following data:

1. Aries Packet Type - A 4-byte unsigned integer specifying the Aries type of this packet. Values greater than zero indicate low-level Aries commands. A value of zero indicates an application-defined packet which will be handled by Voltron rather than Aries. The TSO programmers likely never dealt with Aries commands directly. Valid Aries commands are:

Aries Packet Type	Meaning	Packet Handler
0	Application-specific (handled by Voltron)	Aries_base+0x6e8f
1	AriesClientConnection::InvokeDisconnectCallback	Aries_base+0x6fb4
3	AriesLib_InitializeConnection	Aries_base+0x6ed1
21	(Client->Server only) Client's response to Type 22	n/a
22	Return information about the client's session	Aries_base+0x7249
26	Unknown	Aries_base+0x6fdc
27	PerformanceCounter::ProcessTimingPing	Aries_base+0x7262
28	PerformanceCounter::Initialize	Aries_base+0x7252
29	(Client->Server only) Client's response to Type 28	n/a

31	AriesClientConnection::InvokeRelogonStartCallback	Aries_base+0x6ec5
44	AriesClientConnection::InvokeRelogonCompleteCallback	Aries_base+0x6fa8

2. Timestamp – A 4-byte unsigned integer representing the timestamp for this packet in milliseconds. This timestamp does not correspond to any timezone or epoch. When the client sends packets to the server, it uses GetTickCount to fill in this field, a Windows API function that returns the number of milliseconds that have elapsed since the system was started—however, it does this in a broken way. When the server sends packets to the client, the client does not even look at this field.
    a. The code where the client fills in the timestamp field can be found at Aries_base+0x5630. It appears that the programmers had a preprocessor macro similar to write_dword(dest, src), so they wrote "write_dword(ptr, GetSystemTime());" which expands out into " ptr[0] = (GetSystemTime() >> 0) & 0xFF; ptr[1] = (GetSystemTime() >> 8) & 0xFF; ptr[2] = (GetSystemTime() >> 16) & 0xFF; ptr[3] = (GetSystemTime() >> 24) & 0xFF;" when what they were supposed to write is "temp = GetSystemTime(); write_dword(ptr, temp)".
3. Aries Payload Size – Aries payload size - A 4-byte unsigned integer specifying the size of the payload that follows the Aries header. If there is one Aries frame in the TCP packet, this should be equal to the size of the TCP packet minus 12.

The packet types are summarized below.

Type 1 – Disconnect Packet

This packet causes the client to disconnect from the server.

The payload of this packet is empty.

Type 3 – Defer Connection Packet

This packet causes the client to reconnect to another server without interrupting gameplay.

The format of the payload is as follows:

1. Unknown - 256 bytes containing a null-terminated string
2. Server - 256 bytes containing a null-terminated string in the format "host:port" where "host" is an IPv4 address or a host name
3. Password - 60 bytes containing a null-terminated string corresponding to the "password" field of launchfile.lch

Type 21 – Client Session Info Packet

The client sends this in response to the Type 22 Aries packet.

The format of the payload is as follows, using little-endian integers:

- User - 112 bytes containing a null-terminated string corresponding to the "user" field of launchfile.lch. All data after the string is uninitialized memory (just like Heartbleed).
- AriesClient version - 80 bytes containing a null-terminated string set to "6.0.2.0o"

- Email - 40 bytes containing a null-terminated string corresponding to the "email" field of launchfile.lch
- Authserv - 84 bytes containing a null-terminated string corresponding to the "Authserv" field of launchfile.lch
- Product - A 2-byte unsigned integer corresponding to the "Product" field of launchfile.lch; should be 0x0001
- Unknown - 1 byte set to the character '9' (0x39)
- ServiceIdent - 3 bytes containing a null-terminated string corresponding to the "ServiceIdent" field of launchfile.lch; should be "P"
- Unknown - A 2-byte unsigned integer, typically equal to 1 (after a Type 3 packet) or 4 (before any Type 3 packets)
- Password - A non-null-terminated string corresponding to the "password" field of launchfile.lch; this field has a variable length which should be calculated based on the size of the payload: password_len = payload_size - 331.
- Reserved - 7 bytes of uninitialized memory

Note: The Password field is initially null-terminated when the client reads from launchfile.lch (with the null terminator residing in the first byte of the 7-byte Reserved field); however, when the client receives a Type 3 Aries packet from the city server, the client first copies the password from the Type 3 packet without copying the null-terminator and then subsequently inserts the null-terminator one byte past the end of the string into the second byte of the 7-byte Reserved Data field. (See Aries_base+0x73f9.) Hence, the city server should not assume that the Password field is null-terminated.

Type 22 – Client Session Info Query Packet

When the server sends the Type 22 Aries packet to the client, the client responds with a Type 21 Aries packet containing information about the client's session.

The payload of this packet is empty.

Type 26 – Error Packet

The format of the payload is as follows:

- Error - A null-terminated string of the form "subject:[error]", where subject is "ARIES OS" or "CADENCE" and error is one of the following:
    - PM1000A, PM1001E, PM1002A, PM1003F, PM1004A, PM1005F, PM1006F, PM1007F, PM1008E, PM1009A, PM1010F, PM1011F, PM1012A, PM1013E, PM1014E, PM1015E, PM1016A, PM1017A, PM1018E, PM1019A, PM1020F, AU1000F, AU1001E, AU1002E, AU1003F, AU1004F, AU1005A, AU1006E, AU1007A, AU1008E, AU1009F, AU1010F, AU1011E, AU1012E.

The client does not close the connection, or show any kind of error, in response to this packet. *Perhaps the Error field is really a "last error".*

Type 27 – Timestamp Packet

The format of the payload is as follows:

- Timestamp - A 4-byte unsigned integer specifying the server's prediction of the client's clock once the client receives this packet (in milliseconds)

Type 28 – Latency Test

The format of the payload is as follows:

- Histogram width - A 4-byte unsigned integer specifying the width of the histogram, that is, the highest prediction error that can be displayed on the histogram (larger errors will be clamped); must be nonzero
- Unknown - A 4-byte unsigned integer; must be nonzero (e.g. 1)
- Unknown - A 4-byte unsigned integer; must be nonzero (e.g. 1)

The Type 27 and 28 Aries packets can be used to measure the variance in one-way latency. For an explanation, see: http://niotso.org/files/aries_ping_packets.txt

It turns out that this timing test is not useful for finding the latency itself. To determine round-trip latency, the server should simply use the Type 22 packet as a "ping" packet: that is, find the current time (call it t0), send a Type 22 packet, wait for a response, and find the new current time (call it t1). The round-trip latency is given by t1-t0.

Type 29 – Latency Test Results

The format is as follows, using little-endian integers:

- Unknown - 88 bytes set to zero
- Histogram - 84 bytes (A 4-byte unsigned integer for all 21 bins in the histogram)
- Total clock drift - A 4-byte unsigned integer specifying the sum of the clock drift over all tests

Type 31 – Relogon Start

The payload of this packet is empty.

The relogon start callback function is null (as can be observed by the "test eax, eax" check at Aries_base+0x7cf7), hence the client does not do anything in response to this packet.

Type 44 – Relogon Complete

The payload of this packet is empty.

The relogon complete callback function is null (as can be observed by the "test eax, eax" check at Aries_base+0x7d67), hence the client does not do anything in response to this packet.

Voltron Packets

Voltron Packets are packets designed specifically for The Sims Online and perhaps similar games built on top of similar technology.

These are as they are in The Sims Online: Pre-Alpha.

BC Version List PDU – 0x92

Unknown at this time what this does.

Type	Bytes	Description
String		Version String
String		Str1 – Unknown
uint	1	Arg1 – Unknown

Client Bye PDU – 0x06

Sent when the client is disconnecting.

Type	Bytes	Description
String		Message
uint	1	Status Code

Client Online PDU – 0x09

Sent when the client receives the Host Online PDU.

Contains information about the current game client and user.

Type	Bytes	Description
byte	1	Major version
byte	1	Minor version
byte	1	Point version
byte	1	Art version
uint	4	Arg1 – Unknown
DateTime	4	Time
uint	4	Number of Attempts to Connect
uint	4	Last Exit Code
byte	1	Last Failure Type
byte	1	Number of Failures
byte	1	Is Running – Unknown what this does
byte	1	Is Relogging
byte	1	Arg2 – Unknown

DB Request Wrapper PDU – 0x82

Sent when doing Database Inquiries/Responses and the body contains a packet with Database data.

Type	Bytes	Description
string		Aries ID
string		Master ID
ushort	2	Bitfield_Arg1 – Unknown
uint	4	Message Size `The distance (in bytes) from the end of the DWORD to the end of the packet. Basically, all other fields after this one are included in the "Body" of the packet. For clarity and usability, values stored in the Body have been pulled-up to the class level, such as TSOPacketFormatCLSID`
uint	4	TSOPacketFormatCLSID `TSO has different classes in the library that correspond with the structure of these requests. This is the identifier for which class should be created to house the data.`
uint	4	Transaction ID – An ID given for the current Database Transaction
uint	4	Flags
uint	4	TSOSubMsgCLSID `Beneath the overall packet type there is a CLSID for the individual request being made.`
Byte[]		The body of the Database request.

Packet Request/Response CLSIDs

This request also has CLSIDs which dictate the type of Request/Response is being sent/received. They are as follows:

CLSID	Name
0xFD3338E9	GetRoommateInfoByLotIDRequest
0xDD3339EE	GetRoommateInfoByLotIDResponse
0x5BB73FAB	GetCharBlobByIDRequest
0x5BB73FE4	GetCharBlobByIDResponse
0x7BAE5079	GetCharByIDRequest
0x3BF96A6C	GetRelationshipsByIDRequest
0x5BEEB701	GetLotListRequest
0xFBE96AA3	GetLotByIDRequest
0xDD909124	GetHouseLeaderByLotID
0x5BB8D069	GetHouseBlobByIDRequest
0xFD8F9080	GetBookmarksRequest
0x3D8F9003	GetBookmarksResponse
0x3D03D5F7	InsertGenericLog_Request
0x9BB8EAC4	InsertNewCharBlob_Request
0x1BB8EB44	InsertNewCharBlob_Response

TSO Host Online PDU – 0x1D

Sent to the Client when initiating a connection. Tells the client what version the host is using and the size of packet it is expecting.

Type	Bytes	Description
uint	4	Host Version – Usually 0x0C for Pre-Alpha
ushort	2	Size Limit – Safe bet is to use 1MB here (1024 bytes)
uint	4	Arg3 – Unknown

The Anatomy of a CharBlob

A CharBlob (or Avatar File) is the way that The Sims Online: Pre-Alpha stores and transmits the appearance, personality, name and description, etc. of an avatar.

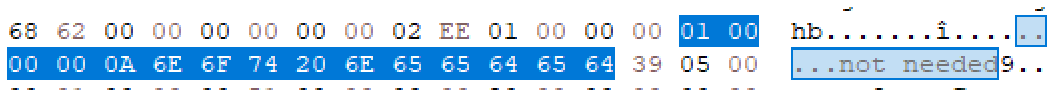
Their traits are applied to a template person known internally as “TemplatePerson” with an object name of “netuser00000” where the number increments up once every time more than one are loaded at a time.

Avatar1.dat

In the UserData directory, a file called Avatar1.dat is created which contains a local copy of the avatar the player owns. It is deleted every time GetCharBlobByID_Response is sent from the server. It is created whenever:

1. A CharBlob is sent to the Client that is successfully extracted and loaded.
2. A new Avatar is created in Create-A-Sim

It’s structure is a RAS_ stream whose payload is the CharBlob itself. A CharBlob starts with the following bytes:



```
68 62 00 00 00 00 00 00 02 EE 01 00 00 00 01 00   hb.......î......
00 00 0A 6E 6F 74 20 6E 65 65 64 65 64 39 05 00   ...not needed9..
```

The AvatarID is encoded into the CharBlob at 0x3D and is stored in Little Endian format. (0x0539 is my AvatarID in this case)

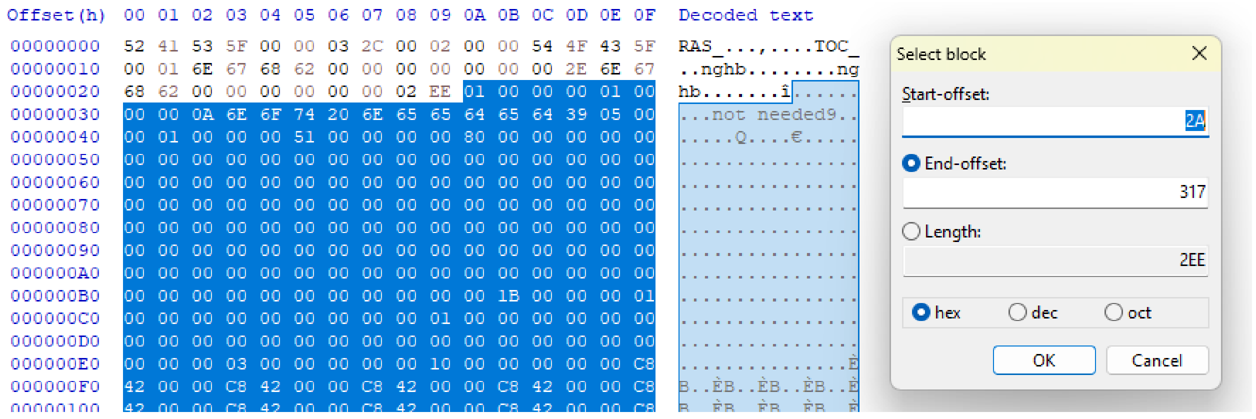
In order for this file to be valid, the AvatarID must match what the server reports is YOUR AvatarID.

You can “transmutate” any CharBlob you want into this file, simply ensure your CharBlob has been:

1. Successfully decompressed from its original RefPack bitstream
2. AvatarID has been changed to that of your own
    a. Note you may also need to change the name of the Avatar

After that is complete, you can copy all the bytes of the CharBlob into the Avatar1.dat file at offset: 0x2E.

You must ensure to change the Little Endian UInt32 stored here to be the length (in bytes) of your CharBlob plus four extra bytes. *In this case the length is 0x02EE*



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

00000000  52 41 53 5F 00 00 03 2C 00 02 00 00 54 4F 43 5F  RAS_...,....TOC_
00000010  00 01 6E 67 68 62 00 00 00 00 00 00 00 2E 6E 67  ..nghb........ng
00000020  68 62 00 00 00 00 00 00 02 EE 01 00 00 00 01 00  hb.......î......
00000030  00 00 0A 6E 6F 74 20 6E 65 65 64 65 64 39 05 00  ...not needed9..
00000040  00 01 00 00 00 51 00 00 00 00 80 00 00 00 00 00  .....Q....€.....
00000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000B0  00 00 00 00 00 00 00 00 00 00 00 1B 00 00 00 01  ................
000000C0  00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00  ................
000000D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000E0  00 00 00 03 00 00 00 00 00 10 00 00 00 00 00 C8  ...............È
000000F0  42 00 00 C8 42 00 00 C8 42 00 00 C8 42 00 00 C8  B..ÈB..ÈB..ÈB..È
00000100  42 00 00 C8 42 00 00 C8 42 00 00 C8 42 00 00 C8  B..ÈB..ÈB..ÈB..È
```

Select block

Start-offset: 2A

[✓ End-offset:] 317

Length: 2EE

[✓ hex] dec oct

CharBlobs from InsertCharBlobByID_Request Payload

These are full CharBlobs created by the Client sent to the Server compressed using RefPack.

However, once decompressed you'll notice the AvatarID field has been set to 0x1111FFFF which is obviously not the AvatarID of the character you just created (most likely)

In this case, before sending back to the Client, you will need to ensure is set to the appropriate value and re-compressed back into a RefPack stream.

HouseBlob seems to start at offset 0x4C from the RAS_ stream the game sends

Create-A-Sim and Creating an Avatar on the Server

Creating an Avatar is the first step to getting to the Map View in The Sims Online: Pre-Alpha. When you first log in, you are greeted with Select-A-Sim with no Sims yet created.

You must create your Avatar now.

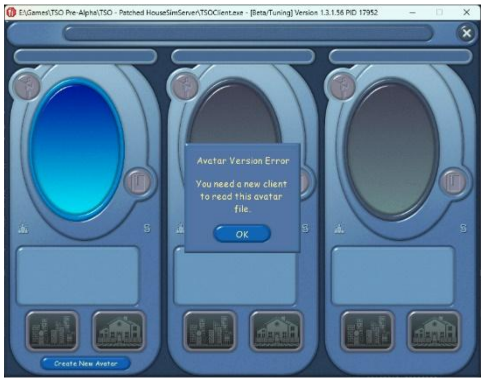
Getting to Create-A-Sim

Getting to Create a Sim requires a connection to TSOHTTPS and TSOTCP. The game will connect to the Shard Selector Servlet with a PlayerID of 0.

The client will then connect to the Shard at the given Connection Address. Create A Sim appears after the TSO Host Online PDU is sent to the game client.

Submitting a New Sim

After hitting “Go to the City” the game will use the connection to TSOTCP to send a DBRequestWrapperPDU with the InsertCharBlob_Request ActionCLSID. The payload being a cTSOSerializableStream with a payload of a Character Blob compressed using RefPack.


E:\Games\TSO Pre-Alpha\TSO - Patched HouseSimServer\TSOClient.exe - [Beta/Tuning] Version 1.3.1.56 PID 17952

Avatar Version Error

You need a new client to read this avatar file.

While the entire structure of a CharBlob (known internally in the game as an Avatar File) is not entirely known, the game will prepare a fully functional CharBlob for us which is sent compressed in the InsertCharBlob_Request payload.

How Select-A-Sim (and the Person Control) works

The game will store a local version of the avatar the player owns in the UserData folder. This is stored as Avatar1.dat. It is a RAS_ stream that holds information on the appearance of the avatar, its personality, Name and Description, and potentially more as well. Included in this stream is also the ID of the Avatar, which must match the ID given by the server for the character to successfully appear in CAS.

The Person Control is the floating 3D render of the avatar that is seen on Select a Sim, Create a Sim, the Person Page and the Map View Gizmo. Whenever a CharBlob is loaded to be previewed in this control is when error checking for the AvatarID has been witnessed.

Loading into Your House

Updated 7/2/2025

Loading into Your House in The Sims Online: Pre-Alpha is documented in nio2so, as follows below. You must follow this procedure to get the Client into a hosting mode (known internally as the HostProtocol)

HostProtocol and VisitorProtocol

The HostProtocol and VisitorProtocol work in tandem to deliver important functionality to the Online gameplay experience in The Sims Online: Pre-Alpha. The HostProtocol handles outbound traffic to the City Server, which acts as the TSOVoltron server.

Voltron is the online gameplay framework powering the MatchMaker (MapView) which allows you to find other online rooms to play in, create your own, etc. It also allows your client to Host your house for others to play on. It does so by designating your client as the HouseLeader. This is designated in the UpdateRoomPDU and also by the GetHouseLeaderByID_Query message PDU. The game uses this information to determine who the host is. If the client joining is the host of the game, they will also be sent the HouseSimConstraints PDU.

The Loading Procedure

These PDUs are sent in this order, with context on each one:

1. When selecting your lot from the MapView, and clicking Join House: LOT ENTRY PDU is sent first.
    a. This PDU contains which house you're trying to load into.
2. If this Client joining is the HouseLeader, which they are because you're joining your own house, the HOUSE SIM CONSTRAINTS PDU is sent next.
    a. This tells the game to load the lot by ID which is enclosed. This ID should match that of the one you're attempting to join.
3. The TSOUpdateRoomPDU is sent next, telling the Client you're now in an Online Room which you should attempt to connect to.
4. **Why is this important?**
    a. The HouseSimConstraints PDU sets the Time Of Day and sends kLoadHouse message to the SimsHandler.
    b. The SimsHandler is enabled at Application Startup, receives the HouseID to load.
    c. This will put the SimsHandler into the kLoadedEmptyState, waiting for your avatar to arrive.
    d. TSOUpdateRoom PDU will invoke kClientHasConnected to be fired which starts the joining process for yourself.
    e. HOWEVER,
    f. This NEEDS to be done AFTER kLoadHouse is fired because the game will not respond to kClientHasConnected properly if the SimsHandler isn't moved into kLoadedEmptyState.
    g. kClientHasConnected being fired causes the AvatarJoinHandler to be enabled. Then since it is awakened, the game will now respond to the kAvatarID message by accepting the visitor to the House. More on this below.

5. The HostProtocol will send kRequestAvatarID next, since kClientHasConnected event has fired.
6. Voltron sends this PDU back to the Client.
7. The VisitorProtocol responds to this PDU with the kAvatarID PDU.
8. Voltron sends this PDU back to the Client.
9. The HostProtocol accepts the kAvatarID PDU, adds this to the Voltron Session ID map.
10. The AvatarJoinHandler should now transition to the kAvatarJoinDone state.
11. The LoadCharacterAppointment should now be invoked automatically by the Visitor Protocol.
12. Your avatar appears on the lot.

The game is now loaded, you are free to interact with your lot as you see fit.

Notes & Archives from Niotso

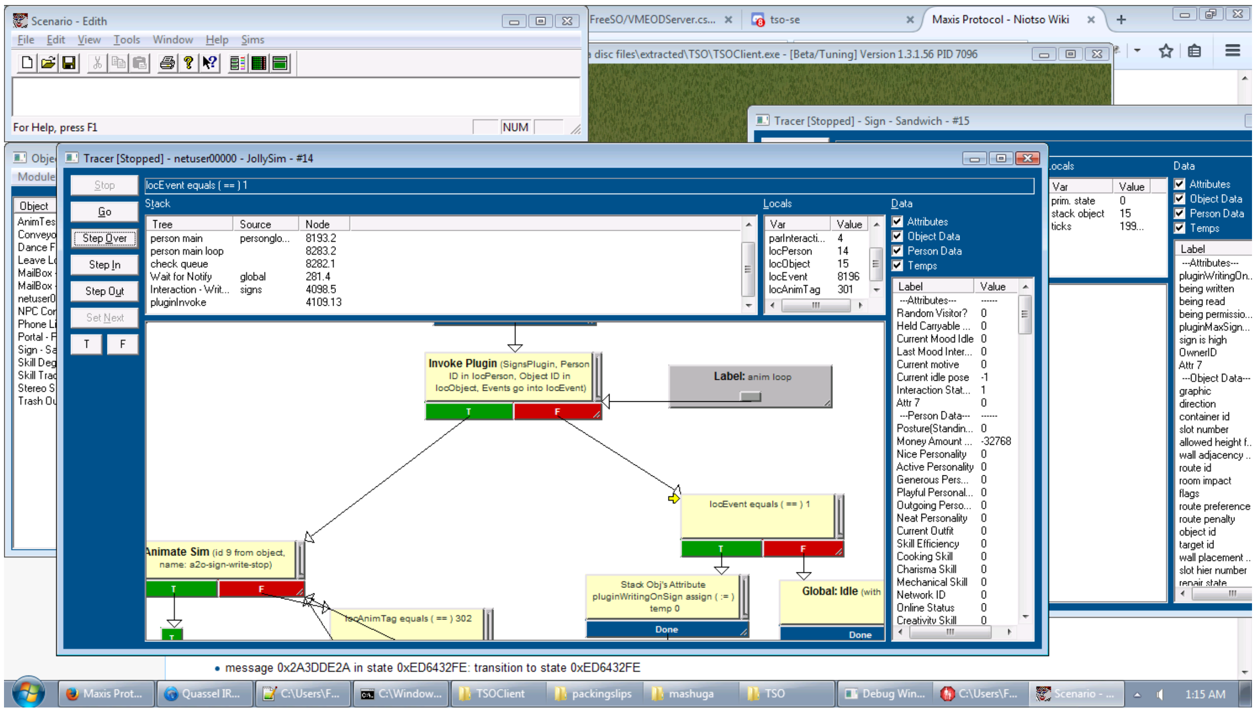
In an effort to preserve the work of niotso, the following notes taken directly from its creator have been copied here for use into the future in case the original source is ever lost/deleted.

Table Found at TSOProtocolsD_base+0x5b9b8

Here is an archive of the TSO Constants Table found on niotso.org. View document.

Image of Pre-Alpha Edith Tool

This is a screencap of the Edith tool running under TSO: Pre-Alpha in a HouseSimServer client.


Scenario - Edith
File Edit View Tools Window Help Sims
For Help, press F1
NUM
FreeSO/VMEODServer.cs...
tso-se
Maxis Protocol - Niotso Wiki
disc files\extracted\TSO\TSOClient.exe - [Beta/Tuning] Version 1.3.1.56 PID 7096
Tracer [Stopped] - Sign - Sandwich - #15
Tracer [Stopped] - netuser00000 - JollySim - #14
Stop
Go
Step Over
Step In
Step Out
Set Next
T
F
Invoke Plugin (SignsPlugin, Person ID in locPerson, Object ID in locObject, Events go into locEvent)
Label: anim loop
locEvent equals (==) 1
Animate Sim (id 9 from object, name: a2o-sign-write-stop)
Stack Obj's Attribute pluginWritingOnSign assign (:=) temp 0
Global: Idle (with
locAnimTag equals (==) 302
Done
Maxis Prot...
Quassel IR...
C:\Users\F...
C:\Window...
TSOClient
packingslips
mashuga
TSO
Debug Win...
Scenario - ...
1:15 AM

locEvent equals (==) 1

Stack

Tree	Source	Node
person main	personglo...	8193.2
person main loop		8283.2
check queue		8282.1
Wait for Notify	global	281.4
Interaction - Writ...	signs	4098.5
pluginInvoke		4109.13

Locals

Var	Value
parInteracti...	4
locPerson	14
locObject	15
locEvent	8196
locAnimTag	301

Data

☑ Attributes
☑ Object Data
☑ Person Data
☑ Temps

Label	Value
---Attributes---	------
Random Visitor?	0
Held Carryable ...	0
Current Mood Idle	0
Last Mood Inter...	0
Current motive	0
Current idle pose	-1
Interaction Stat...	1
Attr 7	0
---Person Data---	------
Posture(Standin...	0
Money Amount ...	-32768
Nice Personality	0
Active Personality	0
Generous Pers...	0
Playful Personal...	0
Outgoing Perso...	0
Neat Personality	0
Current Outfit	0
Skill Efficiency	0
Cooking Skill	0
Charisma Skill	0
Mechanical Skill	0
Network ID	0
Online Status	0
Creativity Skill	0

Locals

Var	Value
prim. state	0
stack object	15
ticks	199...

Data

☑ Attributes
☑ Object Data
☑ Person Data
☑ Temps

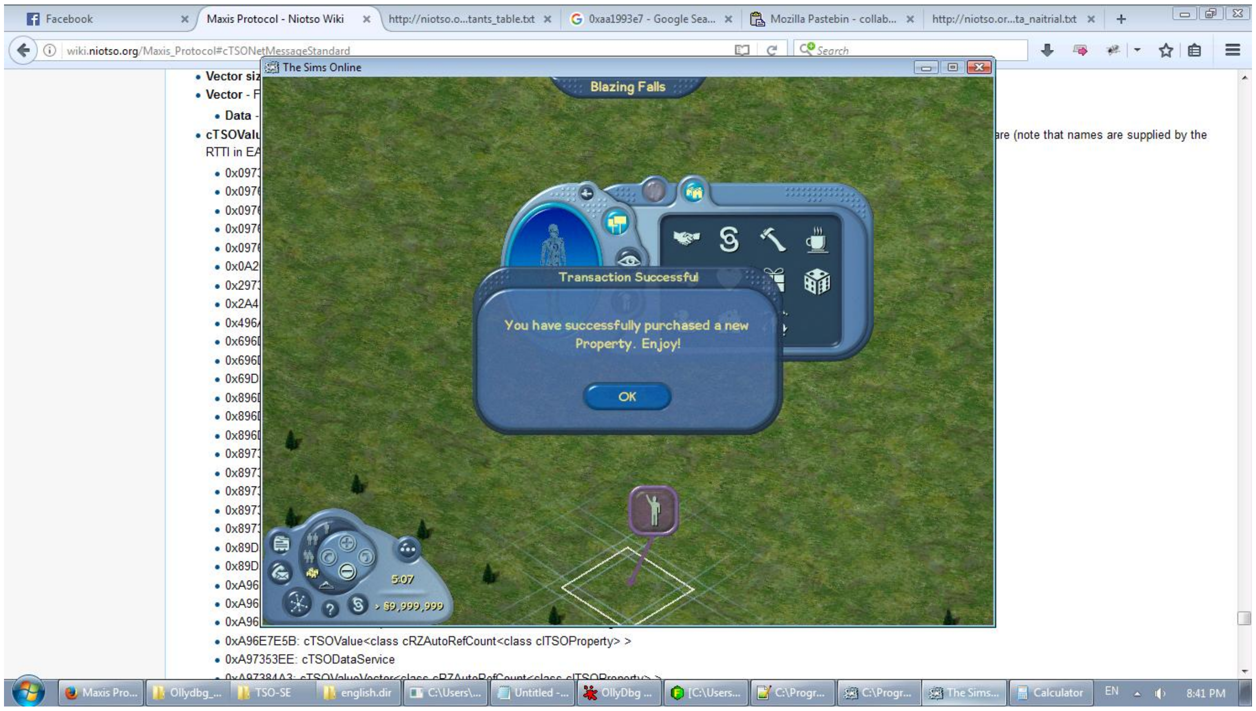
Label
---Attributes---
pluginWritingOn...
being written
being read
being permissio...
pluginMaxSign...
sign is high
OwnerID
Attr 7
---Object Data---
graphic
direction
container id
slot number
allowed height f...
wall adjacency ..
route id
room impact
flags
route preference
route penalty
object id
target id
wall placement ..
slot hier number
repair state

Object
AnimTes
Conveyo
Dance F
Leave L
MailBox
MailBox
netuser0
NPC Co
Phone L
Portal - F
Sign - Sa
Skill Deg
Skill Trai
Stereo S
Trash Ou

• message 0x2A3DDE2A in state 0xED6432FE: transition to state 0xED6432FE

Purchasing a Lot

This is from N&I Trial Version



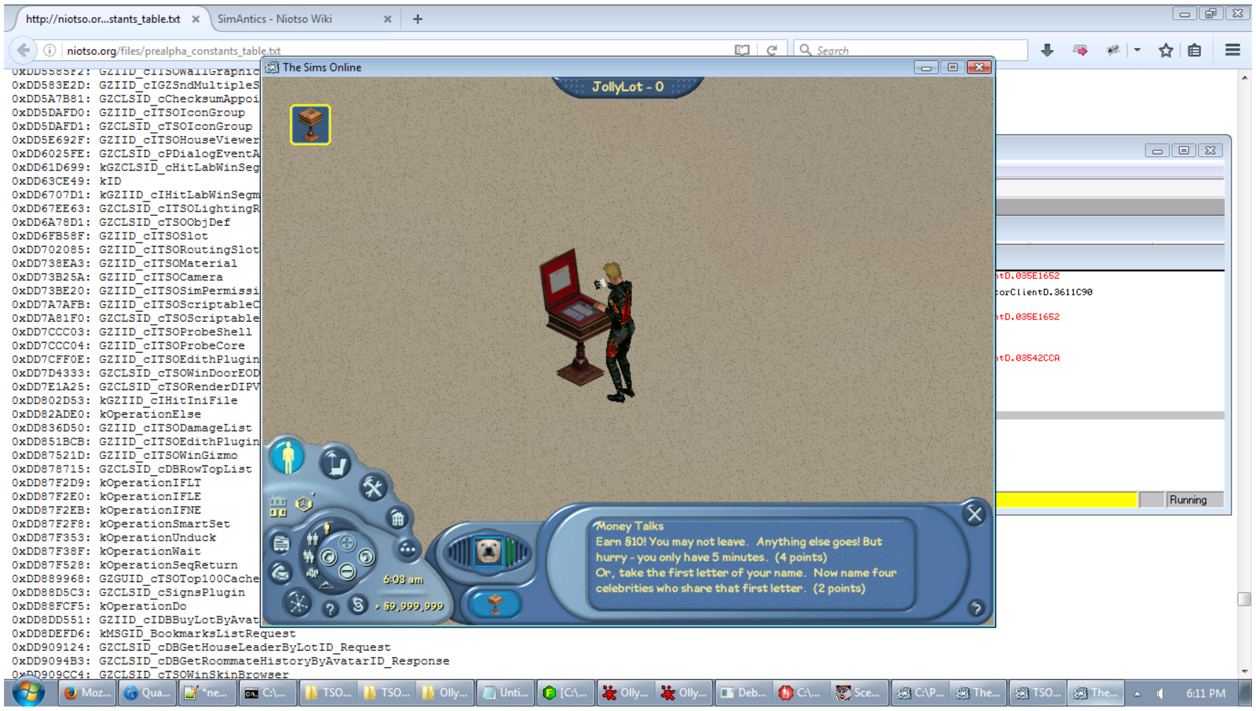
Transaction Successful

You have successfully purchased a new Property. Enjoy!

```
• 0xA96E7E5B: cTSOValue<class cRZAutoRefCount<class cITSOProperty> >
• 0xA97353EE: cTSODataService
```

Using the Sign EOD

This is from N&I Trial Version

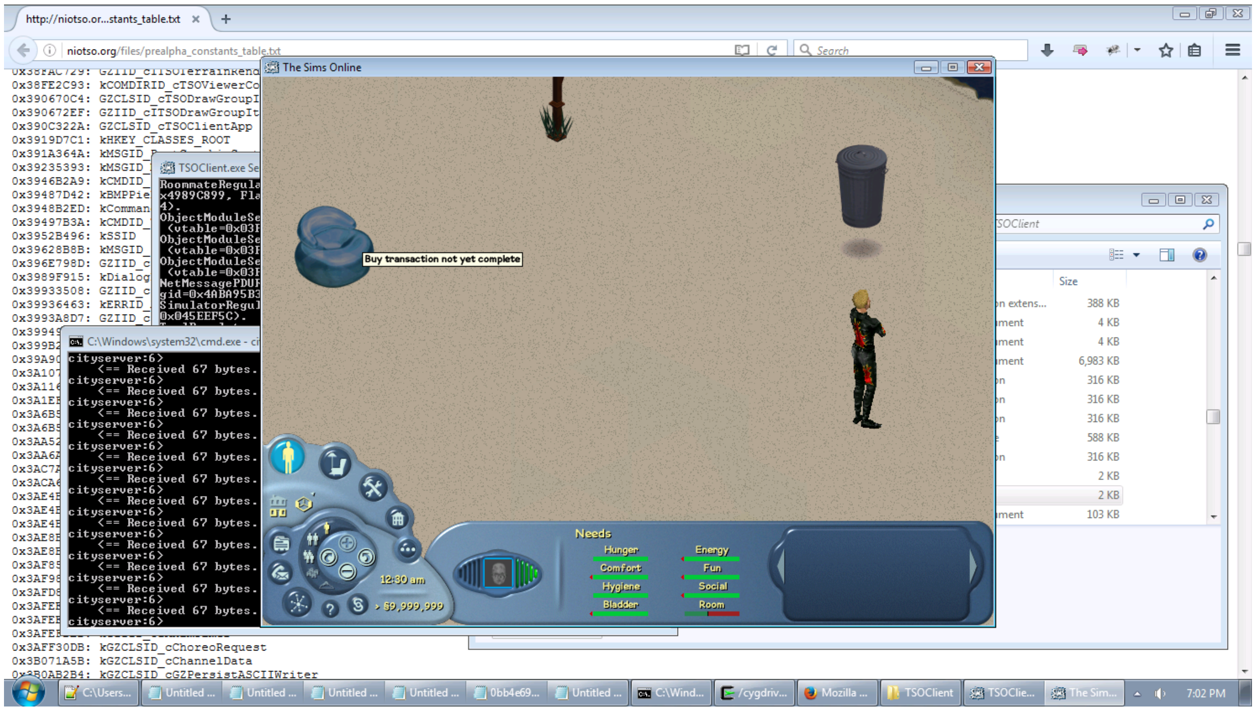


```
0xDD583E2D: GZIID_cIGZSndMultipleS
0xDD5A7B81: GZCLSID_cChecksumAppoi
0xDD5DAFD0: GZIID_cITSOIconGroup
0xDD5DAFD1: GZCLSID_cTSOIconGroup
0xDD5E692F: GZIID_cITSOHouseViewer
0xDD6025FE: GZCLSID_cPDialogEventA
0xDD61D699: kGZCLSID_cHitLabWinSeg
0xDD63CE49: kID
0xDD6707D1: kGZIID_cIHitLabWinSegm
0xDD67EE63: GZCLSID_cITSOLightingR
0xDD6A78D1: GZCLSID_cTSOObjDef
0xDD6FB58F: GZIID_cITSOSlot
0xDD702085: GZIID_cITSORoutingSlot
0xDD738EA3: GZIID_cITSOMaterial
0xDD73B25A: GZIID_cITSOCamera
0xDD73BE20: GZIID_cITSOSimPermissi
0xDD7A7AFB: GZIID_cITSOScriptableC
0xDD7A81F0: GZCLSID_cTSOScriptable
0xDD7CCC03: GZIID_cITSOProbeShell
0xDD7CCC04: GZIID_cITSOProbeCore
0xDD7CFF0E: GZIID_cITSOEdithPlugin
0xDD7D4333: GZCLSID_cTSOWinDoorEOD
0xDD7E1A25: GZCLSID_cTSORenderDIPV
0xDD802D53: kGZIID_cIHitIniFile
0xDD82ADE0: kOperationElse
0xDD836D50: GZIID_cITSODamageList
0xDD851BCB: GZIID_cITSOEdithPlugin
0xDD87521D: GZIID_cITSOWinGizmo
0xDD878715: GZCLSID_cDBRowTopList
0xDD87F2D9: kOperationIFLT
0xDD87F2E0: kOperationIFLE
0xDD87F2EB: kOperationIFNE
0xDD87F2F8: kOperationSmartSet
0xDD87F353: kOperationUnduck
0xDD87F38F: kOperationWait
0xDD87F528: kOperationSeqReturn
0xDD889968: GZGUID_cTSOTop100Cache
0xDD88D5C3: GZCLSID_cSignsPlugin
0xDD88FCF5: kOperationDo
0xDD8DD551: GZIID_cIDBBuyLotByAvat
0xDD8DEFD6: kMSGID_BookmarksListRequest
0xDD909124: GZCLSID_cDBGetHouseLeaderByLotID_Request
0xDD9094B3: GZCLSID_cDBGetRoommateHistoryByAvatarID_Response
0xDD909CC4: GZCLSID_cTSOWinSkinBrowser
```

Money Talks
Earn §10! You may not leave. Anything else goes! But hurry - you only have 5 minutes. (4 points)
Or, take the first letter of your name. Now name four celebrities who share that first letter. (2 points)

Buy Transaction Not Complete

This is from N&I Trial Version

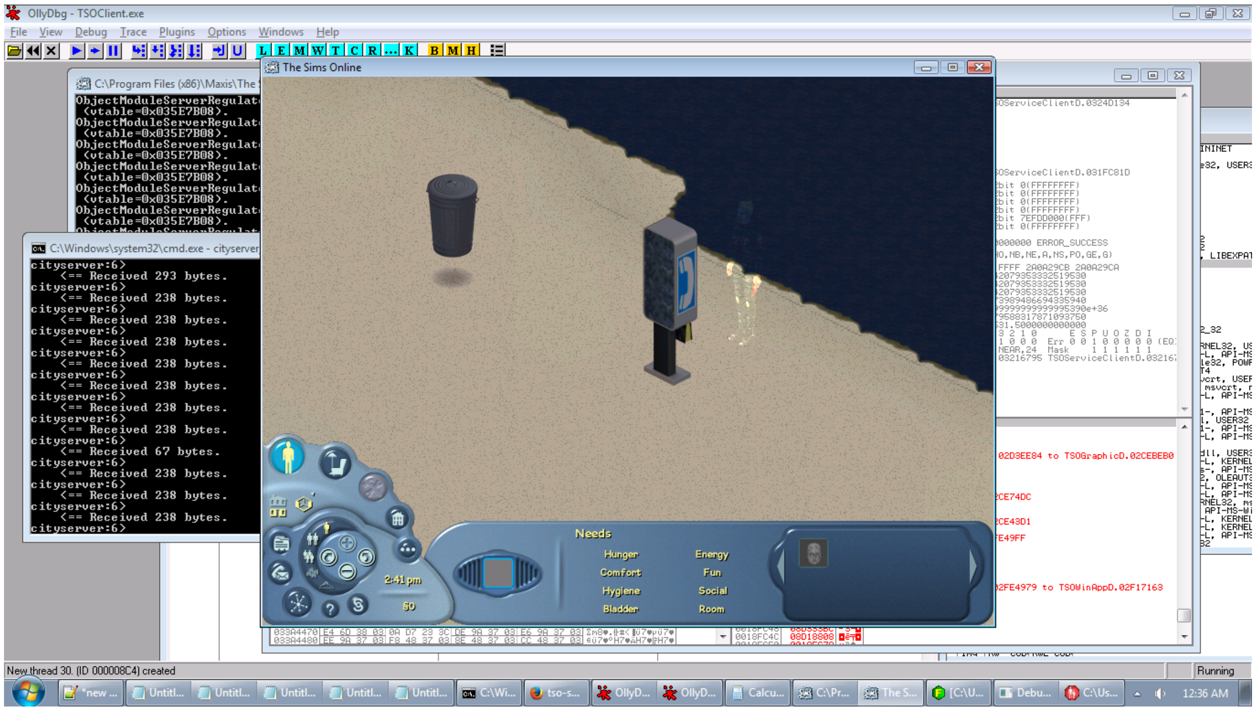


```
0x38FE2C93: kCOMDIRID_cTSOViewerCo
0x390670C4: GZCLSID_cTSODrawGroupI
0x390672EF: GZIID_cITSODrawGroupIt
0x390C322A: GZCLSID_cTSOClientApp
0x3919D7C1: kHKEY_CLASSES_ROOT
...
0x3952B496: kSSID
...
0x3AFF30DB: kGZCLSID_cChoreoRequest
0x3B071A5B: kGZCLSID_cChannelData
```

```
cityserver:6>
     <== Received 67 bytes.
cityserver:6>
     <== Received 67 bytes.
cityserver:6>
     <== Received 67 bytes.
cityserver:6>
     <== Received 67 bytes.
cityserver:6>
     <== Received 67 bytes.
cityserver:6>
     <== Received 67 bytes.
cityserver:6>
     <== Received 67 bytes.
cityserver:6>
     <== Received 67 bytes.
cityserver:6>
     <== Received 67 bytes.
cityserver:6>
     <== Received 67 bytes.
cityserver:6>
     <== Received 67 bytes.
cityserver:6>
     <== Received 67 bytes.
cityserver:6>
```

HouseSimServer Disconnecting

This is from N&I Trial Version



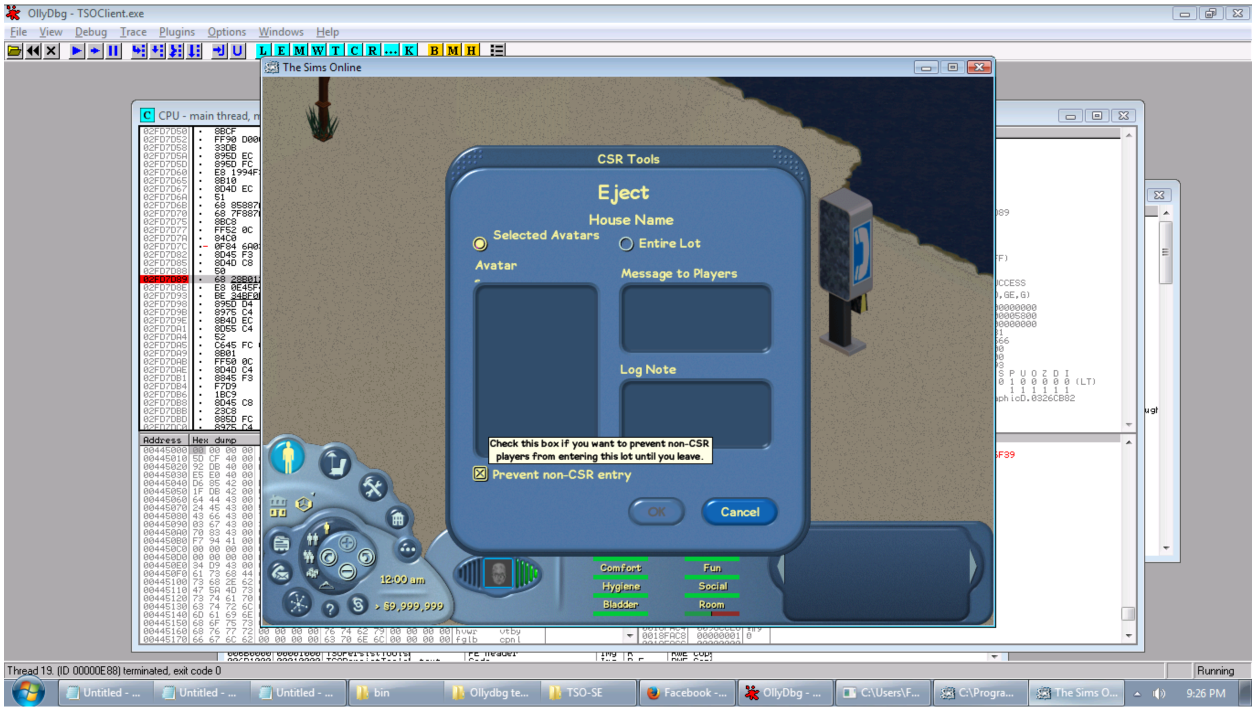
```
ObjectModuleServerRegulat
 <vtable=0x035E7B08>.
ObjectModuleServerRegulat
 <vtable=0x035E7B08>.
ObjectModuleServerRegulat
 <vtable=0x035E7B08>.
ObjectModuleServerRegulat
 <vtable=0x035E7B08>.
ObjectModuleServerRegulat
 <vtable=0x035E7B08>.
ObjectModuleServerRegulat
 <vtable=0x035E7B08>.
```

```
cityserver:6>
     <== Received 293 bytes.
cityserver:6>
     <== Received 238 bytes.
cityserver:6>
     <== Received 238 bytes.
cityserver:6>
     <== Received 238 bytes.
cityserver:6>
     <== Received 238 bytes.
cityserver:6>
     <== Received 238 bytes.
cityserver:6>
     <== Received 238 bytes.
cityserver:6>
     <== Received 238 bytes.
cityserver:6>
     <== Received 67 bytes.
cityserver:6>
     <== Received 238 bytes.
cityserver:6>
     <== Received 238 bytes.
cityserver:6>
     <== Received 238 bytes.
cityserver:6>
```

New thread 30. (ID 000008C4) created

CSR Menu

This is from N&I Trial Version

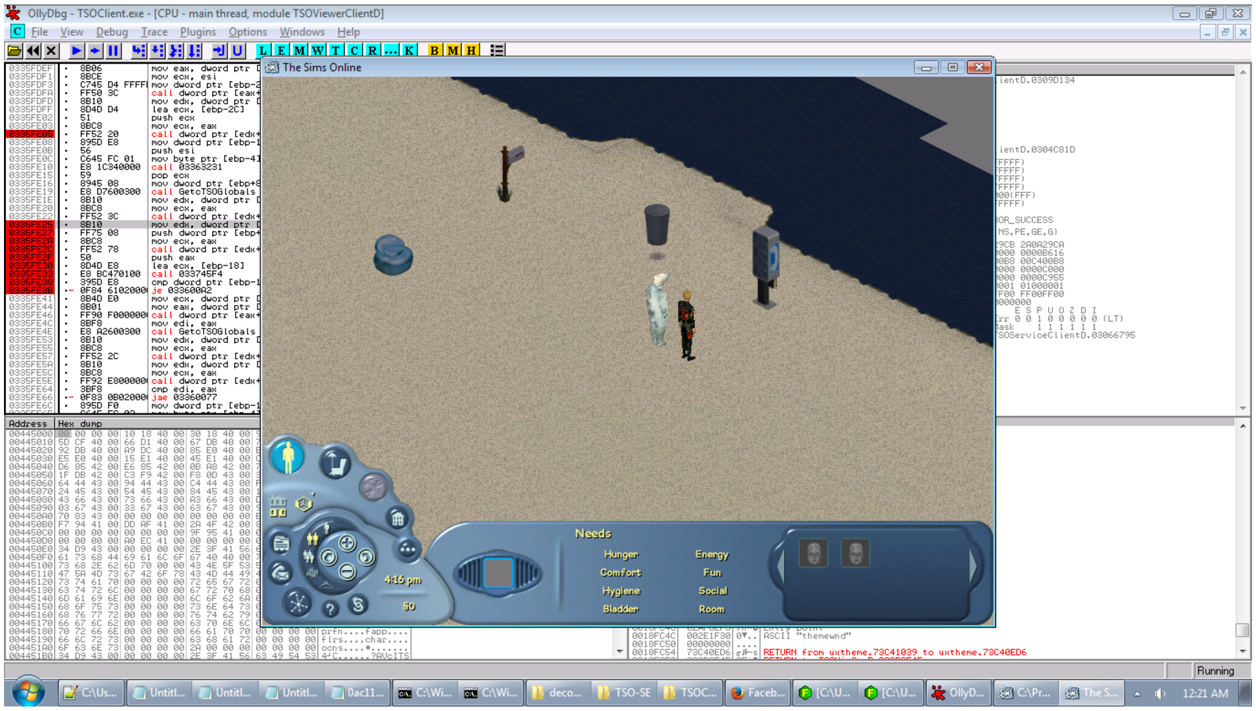


Check this box if you want to prevent non-CSR players from entering this lot until you leave.

Thread 19. (ID 00000E88) terminated, exit code 0

Multiplayer Room

This is from N&I Trial Version


OllyDbg - TSOClient.exe - [CPU - main thread, module TSOViewerClientD]
File View Debug Trace Plugins Options Windows Help
The Sims Online
4:16 pm
50
Needs
Hunger
Energy
Comfort
Fun
Hygiene
Social
Bladder
Room
Running
12:21 AM