

## **Stratotech Grid-wide/Sim-wide Transporter System User Manual**

This user manual has been divided into three sections. The first section is a Quick Start Guide to help you get your system up and running as fast as possible. It is designed for those who hate to read manuals, and don't care how it works, just that it works. The second section covers basic functionality and goes through the Options menus and discusses the various settings available for customizing your transporter network. The third section is a more in-depth discussion of the architecture of the system and instructions and considerations for planning your grid-wide network to get the most out of your system. Also included in the third section are FAQs and a discussion of the APIs available to you for interfacing other scripts to this system.

### **Quick Start Guide:**

1 - Rez the packing crate you received with your purchase and click it to have the contents copied to your inventory.

2 - Place your destination pads or booths wherever you would like them and give each destination a 12 character, or less, name to identify it in the menu system.

Stand on a pad, click it for a menu, and select a destination. If you need only sim-wide operation, these are the only steps required to get a fully functional sim-wide transporter network up and running. The menu system is self-explanatory. However, if you need grid-wide functionality, keep reading...

3 - Repeat Step 2, above, for each sim in your grid-wide network.

4 - Rez a TP Grid Server from the folder you were given when you unpacked your system and place it anywhere you like in the sim. Only one TP Grid Server per sim is required.

5 - Take a COPY of the TP Grid Server you rezzed in Step 4.

6 - Rez a copy of the TP Grid Server you just copied in the next sim, then take a copy of that one to rez at the next sim, and so on.

That's it! Your grid-wide transporter network is now online! But, if you need to Bridge to other transporter networks, keep reading...

7 - To Bridge to another transporter network, click one of the TP Grid Servers you rezzed, above, and select Bridge from the menu and copy and paste the UUID(key) for the avatar who owns the transporter network to which you would like to connect into the Text Box and click Submit. That avatar must also add your UUID to one of their TP Grid Servers, as well. The TP Grid Servers need not be located in the same sim.

8 - Add one URL (http:// address) for one of the TP Grid Servers belonging to the avatar you added in Step 7 by selecting Add URL from the TP Grid Server menu and copying and pasting the URL into the provided Text Box and click Submit. Likewise, provide the other avatar with the URL from one of your TP Grid Servers to be added to their system. You can retrieve the URL for a TP Grid Server by selecting My URL from the TP Grid Server menu.

9 - Repeat Steps 7 and 8, above, for each Bridge you wish to create.

=====

## **Definitions:**

TP Grid Server - This device functions as a node in your grid wide transporter network. It passes along location names, coordinates, and the URLs for each local destination in the region to remote destinations making inquiries from other regions. It also stores network information, such as the URLs of the other nodes in the network and a list of the other avatars whose networks are authorized to communicate with yours. You need one of these devices in each region which you wish to connect to each other across the grid. Management of your transporter network can be accomplished through any of your TP Grid Servers and the changes will be passed along to the others.

URL - Uniform Resource Locator. This is the http:// address assigned to each of the components in your transporter network. Each pad or booth gets its own URL, and so do your TP Grid Servers. These addresses are used for communicating between the devices. We're using URLs to communicate; because, that is the only efficient way objects can communicate with each other across the grid in Second Life. URLs always start with http:// or https:// and should never have spaces inserted in front of them when adding them through Add URL.

Beacon - In this application, a beacon is a URL of which your TP Grid Servers are aware. This does not necessarily mean that your TP Grid Servers are authorized to connect to it, but it does mean that it is being stored and occupying memory in your TP Grid Servers. You can get a list of Beacons of which your TP Grid Server is currently aware by selecting Beacons from the TP Grid Server menu. There should be exactly one beacon for every TP Grid Server in your network, plus one beacon for every TP Grid Server that belongs to other avatars with whom you have established a Bridge. And, there can be other beacons that belong to other avatars with whom you have not established Bridges. These would show up in the list, if they have a Bridge established with an avatar with whom you also have established a Bridge.

UUID - Universally Unique Identifier. This is a unique set of letters and numbers that represent a specific object, avatar, or other resource located in world. In our application, we are using the UUIDs of the avatars we wish to authorize to connect to our transporter network. Other applications of UUIDs in SL include textures and sounds. They are also sometimes referred to as, "keys."

Memory - In this application, this represents the storage capacity of your TP Grid Server to store network information, such as URLs, location names and coordinates, and UUIDs of other avatars whose networks are authorized to connect to yours. You can retrieve the current memory capacity by selecting Memory from the TP Grid Server menu. If memory values fall below 1024 bytes, your TP Grid Server will automatically reset itself to avoid locking up from a "stack heap collision" which basically means that it ran out of storage capacity. If you are running into low memory problems, it is an indication that you may have too many grid enabled destinations in the region, too many remote regions connected together, too many Bridges to networks belonging to other avatars, multiple TP Grid Servers per region, or a combination, thereof. You should seek to reduce the TP Grid Server memory demands by either deleting or making some destinations in the region Local Only, which filters them from the TP Grid Server. You can also reduce the number of regions connected or disconnect from some of your Bridged networks. And, remember, only one TP Grid Server is required per region for your entire network, including those with whom you have bridged your network to theirs. Using more than one TP Grid Server per region will diminish the capacity of your network without providing any benefits.

Bridge - This represents a connection to a transporter network that belongs to another avatar. Bridges are always between the transporter networks of two parties - you and one other avatar. Once a Bridge has been established to another avatar's transporter network, all of their grid enabled destinations become visible to your transporter network and all of your grid enabled destinations become visible to their transporter network. If they have Bridges to other avatars' networks with whom you do not have Bridges established, you will be unable to see their destinations. Although, their TP Grid Server URLs will become visible to your TP Grid Servers and your TP Grid Server URLs will become visible to their TP Grid Servers, even though Bridges have not been established between your network and theirs. That is important to realize for two reasons: 1 - It makes it even easier for you to create a Bridge with them, since you already have their URLs and need only add their UUID to complete the Bridge (they must, of course, add your UUID to their network, as well). And, 2 - Their URLs for their TP Grid Servers WILL take up some of the memory in ALL of your TP Grid Servers, and likewise, your TP Grid Servers will take up some memory in ALL of their TP Grid Servers. Since, memory is a finite resource in your network, it is important to be cognizant of those factors that can affect it.

Beam - in the context of our application, a beam is an object that is rezzed by a pad or booth that starts a particle effect, and is responsible for transporting avatars within a region. A beam is also used to send avatars across the grid to other regions - however, the beam object itself never leaves the region. Another Beam is rezzed at the remote destination to receive the teleporting avatar.

RLV - Restrained Love Viewer. RLV is a method of forcing other avatars to sit, stand, teleport, or otherwise provide a means for one avatar to restrain or force behavior on the part of another avatar. In our application, we are using only two commands: Forced Sit, and Forced Teleport. No other RLV commands are supported by this transporter system. The reason we use RLV is to make teleporting easier. It allows the beam to force us to sit on it to be transported within a region without having to click it, and it allows the beam to force us to teleport to a remote region and coordinates without giving us the grid map or having to click Teleport from the map screen. This feature requires that the avatar be using an RLV compliant viewer. like Firestorm. And, if they do not own the system, they must be wearing an RLV Relay of some kind that allows others to control their avatar with RLV. These are available for free on the Marketplace and your system came with a copy/transfer RLV Relay specifically designed for use with this transporter system and has no other functions. Pass these to your friends and have them wear it on an unused attachment point. It is invisible.

=====

## **Options Menu:**

On/Off - This function turns this particular destination On or Off. When Off, the destination is unavailable for incoming or outgoing transports and will not be displayed in the menu system.

Kiosk On/Off - When set to On, this setting causes the pad or booth to display Local Destinations when clicked. When set to Off, this setting causes the pad or booth to display the main Transport Menu when clicked. Selecting Cancel from the Local Destination menu will bring up the main Transport Menu.

Local On/Off - When set to On, the pad or booth will display only Local Destinations and its name will not be displayed in destination menus in other regions. Transport to other regions is also inhibited.

Colors - This menu allows you to set the Start and End colors for your beams, as well as your hover text color. Wear the Transporter HUD w/Color Change and select which color parameter you wish to adjust. Then, drag across the color pad to select the color you desire. This is a sim-wide setting and affects all of your destinations in the region.

Set Name - This function allows you to set the name of the destination. Destination names should be 12 characters, or less. Add a "!" to the beginning of a destination name to designate it as "Owner Only" access, or add an "\*" to the beginning of a destination name to designate it as "Group Only" access.

Outgoing - This is a toggle On/Off. When On, the destination is available only for outgoing transports and it is not displayed in any destination menus.

Beam Settings - This menu allows you to select your beam style and adjust your beam radius.

Beam Radius - This is the distance in meters from the center of a pad, booth, or HUD user inside which all avatars will be given a beam when a teleport destination has been selected. The HUD defaults to a Beam Radius setting of 1 meter. The Beam Radius defaults to the radius of the pad (minimum 1.75 meters) or 1 meter for the HUD. The maximum Beam Radius setting is 10 meters.

Access - This menu allows you to set all of the destinations in the current region to Public, Group Only, or Owner Only access. When set to Group Only or Owner Only access, none of your destinations will be displayed in the menus of other avatars with this system, unless you have established a Bridge to their transporter network.

HvTxt On/Off - When set to On, the destination name will be displayed in hovertext above the pad.

## Default Settings:

Once a destination pad or booth has been given a name, the Default settings for that individual destination are as follows:

On/Off = On

Local Only On/Off = Off

Outgoing Only = Off

Kiosk Mode = On

HoverText = Off

Beam Radius = 3.25 meters

These settings affect all destinations in the current region:

Beam Style = Dbl Helix

Start Color = <255,255,255> (white)

End Color = <255,255,255>

Access = Public

=====

## Discussion and Network Planning:

The system operates as a stand-alone peer-to-peer http network. It requires no web server or other outside resources for continued operation. The network information is shared between the nodes in the network, providing fault tolerance through script resets and sim restarts, requiring no notecards. This architecture also allows for the management of the network from any TP Grid Server. Let's discuss how the system accomplishes this, and to do this we'll use an example:

John has a transporter network at Seraphim Island. He wants to connect it to four other regions: Willows of Fablewood, New OA, Punta Arena, and Queue.

Kymerlie has transporter networks located at Willows of Fablewood, New OA, and Punta Arena. Life has a transporter network at Queue. None of these avatars have rez rights in the other regions. But, they all want to have access to the destinations belonging to the others. Here's how we set up the network:

1 - Each of the avatars sets up their destinations in their regions and rezzes one TP Grid Server in each region that they own. So, Kymerlie has three TP Grid Servers to rez, but John and Life have only one each to rez. Kymerlie needs to rez one TP Grid Server in her first sim, then take a copy of that TP Grid Server (because it will store the URL for the first one), then rez a copy of that copy at each of the other two sims. She could also take a copy of the second TP Grid Server she rezzed and rez a copy of that at the third sim(because, it will have stored both of the URLs for the other two, already).

2 - Each of the avatars in the new transporter network must add the UUIDs (keys) for the others by selecting Bridge from the TP Grid Server menu and copying and pasting the keys, one at a time, into the text box and click Submit. So, John needs to add UUIDs for Kymerlie and Life. And, Kymerlie needs to add UUIDs for John and Life. And, Life needs to add UUIDs for John and Kymerlie. Of course, Kymerlie need only add the UUIDs into one of her three TP Grid Servers, since the TP Grid Servers will automatically share the new Bridge information with each other. This step establishes the authorization for the networks to communicate with each other.

3 - Each of the avatars should now retrieve the URL for one of their TP Grid Servers by selecting My URL from the menu and provide it to the other avatars in the network. Then, all of the avatars in the network need to add the URLs they got from the others into their TP Grid Servers through the Add URL function in the menu. So, John needs to add a URL from Kymerlie and a URL from Life. Kymerlie needs to add a URL from John's TP Grid Server at Seraphim Island and a URL from Life's TP Grid Server at Queue. This establishes the connections for which we established authorization in step 2.

The example, above, illustrates how you can connect five sims involving three different owners setup from scratch. But, let's say John and Kymerlie already had their Bridge created and Life wants to join the network? It's actually even easier. Life need only add the UUIDs for Kymerlie and John through Bridge, and then add only one URL from either John or Kymerlie through Add URL. And, John and Kymerlie need only add Life's UUID through Bridge in the TP Grid Server menu - and no URL from Life. Why the difference? Because, John and Kymerlie's networks have already shared their URLs with each

other and as one of them picks up the new URL from Life's network (when Life's TP Grid Server connects to it), it will be passed automatically to the others. And, as Life's TP Grid Server connects with either John's or Kymberlie's, it will pick up the other URLs from the other TP Grid Servers automatically.

Let's change the example, above. Suppose John and Kymberlie have already set up a Bridge between their transporter networks so that they can each have access to destinations located at Seraphim Island, New OA, Willows of Fablewood, and Punta Arena. And, Life wants access only to John's destinations at Seraphim Island. In this case, only John and Life need add each other's UUID's through Bridge and Life need only add the URL for John's TP Grid Server at Seraphim Island. Kymberlie need not be involved and she will not see destinations at Queue and Life will not see destinations at New OA, Willows of Fablewood, or Punta Arena. But, because John has established a Bridge with Kymberlie and a Bridge with Life, he can see destinations at all five sims.

These are just a few scenarios that are possible with your grid-wide transporter network. But, they should be sufficient to let you know what kinds of capabilities you now have.

## **To Bridge, or not to Bridge? That is the question:**

And, the answer depends on your overall preferences for your network. Everyone's network is going to be unique and should be designed to fit their needs. So, here are some guidelines to consider in choosing for yourself whether you should Bridge with the networks of other avatars, or not.

First of all, remember that each TP Grid Server added to your network increases the memory requirements for the network. This is true whether the added TP Grid Server represents a new region you're adding, or if it is a TP Grid Server that belongs to another avatar with whom you have created a Bridge. However, additional TP Grid Servers present in your region will have NO EFFECT on the memory requirements of your network, if those TP Grid Servers are not a part of your network. In fact, your destinations will ignore the other TP Grid Servers in the region that do not belong to you - regardless, of whether you have bridged with them, or not.

Also, remember that when you create a Bridge with another avatar's network, it is not necessarily just one TP Grid Server being added to your network - you are adding ALL of the other avatar's TP Grid Servers to your network, and yours are added to their network. The overall size of the network is limited for all of the members of the network. Whether the network is made up of the destinations and TP Grid Servers of one avatar, or a dozen, the total maximum size of the network is the same. So, all of the avatars that are bridged together have made a decision to share the capacity of one network. Read that last sentence, again.

While bridging networks provides some wonderful efficiencies and benefits, clearly, there are some potential drawbacks that must be considered when making the decision.

Bridging allows you to have access to destinations belonging to other avatars in other regions by using their TP Grid Server in their region. This is a perfect solution for you when you do not have rez rights in the remote region - but, the other avatar does. If you have no ability to rez in a remote region, the only way you are going to have access to their destinations from yours across grid is to create a Bridge with that avatar's transporter network. This one benefit, alone, may make it worth it for you to incur the extra memory consumption that a Bridge represents - you decide.

Just be aware of how much extra memory the Bridge might consume - because, every extra TP Grid Server added consumes a little (about 182 bytes). If there is only one TP Grid Server involved, the issue is simple. But, not so simple, if the other avatar has an extensive network that spans 50 regions, and you have only a few. Do you really want your own personal network limited by the size of someone else's network? And, do you want to be subject to the decisions others make about adding other regions or other Bridges to a network with which your network is a part? You decide.

The simple alternative is for each avatar to have one TP Grid Server per region that belongs to them, and not have a bridge at all. This allows each avatar's network to be exactly the size and complexity they want it to be without having to consider anyone else's networks in their decisions. This will allow each avatar that has a TP Grid Server to have access to all Public destinations within the region, regardless of who owns them. But, it does, of course, require that the avatars have rez rights in the regions.

=====

## **Frequently Asked Questions:**

Q: I click the pad. I select my destination. But, I'm not getting a beam. Why not?

A: Many teleporters in Second Life allow the user to click the pad from some distance away from it and select their destination. Then, the pad rezzes a beam that the user clicks to sit and the beam carries them to their destination. That works, but the effect is less than "authentic" from a role play standpoint (if you're into Star Trek role play, anyway). So, this system is different. It requires all those who wish to transport to be standing on the pad, or within the set Beam Radius distance from the center of the pad or HUD user to get a beam. And, if you are using RLV, you need not even click the beam to transport. You will automatically be seated on it and sent along to your destination.

Q: How many destinations can I have with this system?

A: That depends on the design of your transporter network. Your TP Grid Server maintains various kinds of data needed to keep your network running. It stores all of the local destination names, their coordinates in the sim, and their URLs. It also stores the URLs of the other TP Grid Servers in the network along with the names of the regions they serve, both yours and those belonging to other avatars with whom you have created Bridges. And, it stores your list of authorized networks. All of this

information must fit into the memory for one script. A single script can use up to 65,536 bytes of memory. But, a script not only contains data, it also contains the code for the script. And, there are limits to how much data can be transmitted from one TP Grid Server to another - 16,384 bytes at a time.

What this means is that the more regions you have connected and the more Bridges you have established, the fewer destinations you can have per region. And, the more grid-enabled destinations you have in a particular region, the fewer other regions and Bridges you can have in your network. But, to put this into perspective, this system should allow you to have hundreds of destinations in your network. For example, you could conceivably have 20 destinations in each of 20 different regions and still have memory to spare. The test bed for developing this system involved five sims ranging from 1 to 45 destinations each and maintained three Bridges. Alternately, you could probably have as many as 50 regions connected with eight to ten destinations in each. But, these figures are based on loose calculations. Your actual maximum number of destinations is going to be based on the total design of your network.

Q: If I use all of my transporter network's memory, can I just add another TP Grid Server per region to extend its memory capacity?

A: No, the problem with that is that both TP Grid Servers in each region would be maintaining the same data - there would just be two copies of it. In fact, adding an extra TP Grid Server would increase the memory load on the system, because each TP Grid Server has its own URL that must be stored by all of the other TP Grid Servers in the network.

Q: If I establish a Bridge with another avatar's network that also has destinations in the same region as my destinations, whose TP Grid Server should we use? We only need one, right?

A: Both. Each avatar who has a transporter network in the region should have one TP Grid Server of their own to provide access to that avatar's entire transporter network, including their other Bridged networks. Remember, though, that each additional TP Grid Server you have in your network takes up some memory and having multiple TP Grid Servers in the same region will diminish the overall capacity of your transporter network. Pro Tip: If an avatar has only one Bridge established to another avatar's network, and there are no destinations of their own in the remote regions, there is only a need for one TP Grid Server per region in the Bridged network.

Q: Can I have access to my destinations in other regions through a TP Grid Server that I do not own?

A: Yes, as long as you have established a Bridge to that avatar's transporter network and you are using one of their pads or booths. Your pads and booths listen only for your TP Grid Servers and ignore all others in the same region.

Q: Another avatar would like to Bridge with my network to see my destinations, but I don't want or need to see theirs. How can we accomplish this?

A: All destinations in the region that have been set to Public and not set to Local Only can be detected by any TP Grid Server. In this case, you might consider allowing them to rez a single TP Grid Server in

your region so that they can have access to the destinations in your region, but not establish a Bridge with them. Since, they will have a TP Grid Server of their own in the region, they will still have access to their entire transporter network through their HUD, or any destinations that they happen to own in the region. Establish a similar arrangement with them for any other regions, as needed.

Q: How does this system remember all of its data, if it does not use notecards or web servers?

A: The data required to construct and maintain the network is copied to all other TP Grid Servers in the network. If a particular TP Grid Server becomes disconnected for some reason, like due to a sim restart or resetting the scripts, the data is retained in the other TP Grid Servers in the network. And, when a disconnected TP Grid Server reconnects, the other TP Grid Servers will send it the network data it needs. Location information is automatically reconstructed when the TP Grid Server pings all of the destinations to have them identify themselves to the network, so it need not be stored. And, location information is gathered in real time when a user requests a list of local destinations from another region.

Q: I just reset the scripts in my TP Grid Server; because, I wanted to create a completely new network. But, the TP Grid Server automatically connected to the URLs for the old network, anyway. Why did it reconnect to the old network? How did it still know the URLs? How do I get it to disconnect from the old network, so I can set up a new one?

A: There is a script in your TP Grid Server, called Network Kickstart Module that uses a compression algorithm by Becky Phippen to store up to six URLs in the non-volatile hovertext memory for the prim in which the script resides. When your TP Grid Server is reset or your sim is restarted, the Grid Server script asks the Network Kickstart Module script for the stored URLs in the hovertext memory so it can re-establish its connections to the transporter network. The hovertext memory is the ideal place to store these URLs since that information is not lost if the scripts in the object are reset, or even deleted. To erase these URLs for your purpose, use the Reset function in the menu of the TP Grid Server. Or, you can rez a fresh TP Grid Server from the folder you received when you made your purchase.

Q: I have a large transporter network and when I add a new TP Grid Server to the network for a new region, the new region can see the destinations in all the other regions really fast. But, the other regions are not able to see the destinations at the new region for awhile. I get regular "Connected to <region name>" messages, but it takes awhile before they are all connected. Why does it take so long for the other regions to be able to access the new destinations at the new sim?

A: A new TP Grid Server is going to get a new URL that is unknown to the rest of the transporter network. The TP Grid Servers share information by working through the list of known URLs and taking the new one and passing it along to the next URL in its list. The next TP Grid Server in turn does the same thing, passing along the new URL to the next URL in its list. This process continues until all the TP Grid Servers are aware of the new URL. If the URL is assigned to a TP Grid Server in a region that is already listed in the Regions list, the old URL will be replaced with the new one. The first time a new TP Grid Server makes a connection with another one of your TP Grid Servers, it will be passed a list of the URLs associated with all of the other TP Grid Servers in your network - so, that is why the new TP Grid Server can see all the destinations in the other regions, while it takes some time for the other regions to

detect the new region's destinations. In any event, this process should never take more than about ten minutes, even in a large network. And, once the process is complete, the connectivity is established for all regions. The process is limited by the number of http requests Linden Laboratories allows per second per object. Requests are throttled at approximately 25 requests per 20 seconds.

Q: I noticed that when I'm teleporting within a sim, I always land right on the pad. But, when I'm teleporting across the grid to another region, I sometimes land a little above the pad or sometimes I get pressed down into the pad. Why am I not landing right on the pad?

A: Sim-wide transports can be calculated and specified to a precision of 0.00001 meters. But, grid-wide transports can only be specified to a precision of 1 meter. So, the destination coordinates are rounded either up or down from the decimal value to an integer value. Therefore, your landing position can vary by as much as 0.5 meters on each axis.

Q: When I'm using my HUD to transport to destinations in other regions, sometimes I get a beam when I leave the original region and sometimes I don't. Why don't I always get a beam?

A: If you are in a region that allows Group Building, but you do not have the assigned group active, the system assumes that you cannot rez in the region and does not make the attempt to rez a beam, rather than make the attempt and possibly generate an error message. This, of course, will happen anytime you attempt to transport to a remote region when you do not have rez rights. The solution is to wear the proper group tag prior to transporting, even if you are the owner of the parcel or the group to which it has been deeded.

Q: How is it that I am able to access my entire transporter network through the HUD from regions where I don't even have a TP Grid Server, but my pads need a TP Grid Server for access to the entire network. Why the difference? Why don't I need a TP Grid Server in the region when I'm using the HUD?

A: Whenever you enter a region containing one of your TP Grid Servers, it will identify itself to your HUD and the HUD will store its URL. It keeps that URL in memory until you enter another region with one of your TP Grid Servers. So, when you go to regions where you have no TP Grid Servers located, your HUD just uses the one at the last region you were that had one. Remember, any of your TP Grid Servers can provide you with access to your entire network, including your bridged destinations belonging to other avatars.

Q: Can I set up two destinations in a region that are connected only to each other and are not accessible from any other destinations, like to take me from downstairs in my skybox to my bedroom upstairs?

A: This system is designed for those who need sim-wide and/or grid-wide travel, and usually to many destinations in each region, not just between two destinations in one region. If you have a need for such a teleporter, there are many inexpensive, or even free, alternatives. While this system will support only two destinations, isolating those two destinations from all the others in the region is not one of its features.

## User APIs:

If you are not a scripter, you can ignore all of this. But, if you are a scripter, you can use this information to customize your transporter system to meet your specific needs.

Turn the pad or booth On or Off by sending these link messages:

```
llMessageLinked(LINK_SET, 0, "tp on", NULL_KEY);
```

```
llMessageLinked(LINK_SET, 0, "tp off", NULL_KEY);
```

Alternately, you can detect when a pad or booth has been turned on or off by listening for these link messages. You could use this API, if you would like to make an On/Off button for other avatars, besides the owner, or perhaps for a centralized sim-wide transporter pad management script to allow you remote access to the pads to turn them on or off.

Use this link message to send the transporter menu to a specific avatar:

```
llMessageLinked(LINK_SET, 0, "transport", id);
```

If Kiosk Mode is set to On, id will get a menu of destinations in the current region. If Kiosk Mode is set to Off, they will get the full transporter menu. Selecting Cancel from the destinations menu will also bring up the full menu. Using this API will cause "Main" to be added to the menu. If the user selects "Main" from the menu, the following link message is generated and can be used by your scripts:

```
llMessageLinked(LINK_SET, 0, "main menu", NULL_KEY);
```

This can allow you to integrate the menus for the transporter system with your own menus in your own scripts. If no other scripts are present to intercept this link message, it otherwise does nothing when the user selects Main from the menu.

Here is a simple script to allow anyone to turn a pad on or off and get the menu when they turn the pad on:

```
integer toggle = 0;
```

```
default
```

```
{
```

```
    state_entry()
```

```
    {
```

```
    }
```

```
    touch_start(integer total_number)
```

```

{
  if (!toggle)
  {
    IIMessageLinked(LINK_SET, 0, "tp on", NULL_KEY);
    IIMessageLinked(LINK_SET, 0, "transport", IIDetectedKey(0));
    toggle = 1;
  }
  else if (toggle)
  {
    IIMessageLinked(LINK_SET, 0, "tp off", NULL_KEY);
    toggle = 0;
  }
}
}

```

Drop the above script into a button prim linked to the pad.

Here is a simple script to transport everyone on the pad to specific coordinates within the region:

//This script will cause the transporter pad to rez a beam for everyone standing on the pad and send them to specific coordinates within the region

vector OFFSET = <3.25,3.25,0.0>;//This is an offset constant that represents the radius of our largest pad and normalizes the center to the SW corner of the pad. Do not change this.

vector landing\_point = <128,128,21>;//We end up here, plus whatever our avatar's actual offset was on the pad.

default

```

{
  state_entry()
  {
    landing_point -= OFFSET;
  }
}

```

```

}

touch_start(integer total_number)

{

    llMessageLinked(LINK_SET, 0, "/warpPos " + (string)landing_point, NULL_KEY);//Note the single
space after warpPos

    llMessageLinked(LINK_SET, 1, "beam_trigger", NULL_KEY);//We send a 1 when we're ready to go,
resets to 0 automatically after transport

    llMessageLinked(LINK_SET, 0, "/rez_beam", NULL_KEY);

}

}

```

The system sends the avatar's offset position on the pad as a start parameter to the beam when it is rezzed. The beam then adjusts its received destination coordinates by the avatar's offset to determine its landing position at the destination. This keeps all the avatars from piling up on top of each other.

Drop this script into a button prim that is linked to the pad. You could use this API to create your own pad-less destinations. Drop a texture on the face of the prim to show the user where they're going.

Use this API to rez a beam for every avatar standing on the pad and transport them to a specific named destination:

```
llMessageLinked(LINK_SET, 0, "/destination " + destination_name, id);//Note the space after
/destination.
```

Where destination\_name is a 12 character or less destination name in the menu system for the local region and id is the key of the user. The menu system will use id to determine if the avatar has permission to transport to the destination and give them a denial message if they are not authorized to access that destination. Likewise, other avatars on the pad will be given a beam and their keys will be tested for access to the destination, as well. You could use this API to create a large teleport board that users could click button prims with textures on the front face that provided a snapshot or more descriptive name for the destination, to circumvent the 12 character name limitation of the menu system.

Caveats: If the destination pad is missing or Off, this API will silently fail. This API will work only sim-wide.

Here's a sample button prim script:

```
//This script will cause the transporter pad to rez a beam for everyone standing on the pad and send them to the named destination
```

```
string destination_name;
```

```
default
```

```
{
```

```
    state_entry()
```

```
    {
```

```
        destination_name = llGetObjectNeame();//Put your 12 character or less destination name in the Name field for the button prim
```

```
    }
```

```
    touch_start(integer total_number)
```

```
    {
```

```
        llMessageLinked(LINK_SET, 0, "/destination " + destination_name, llDetectedKey(0));
```

```
    }
```

```
}
```

**IMPORTANT:** User scripts and other contents placed in the pad prims are subject to being removed when the pads are updated to a later version. Where possible, locate your own scripts and content in other prims than the prim containing the transporter system scripts.

Using your own Transporter Pad objects:

You are welcome to move the transporter system scripts into objects of your own creation and design. To that end, you should study the full perm TP on/off listener script that is in the pad. You will see that you can have certain prim faces be made fullbright or add glow or adjust other primitive parameters in response to link messages from the system. And, remember to remove the TP on/off listener script in the pad prim, if you are using one that accomplishes the same task, to avoid potential conflicts between the scripts.

You will see in the TP on/off listener script that your pad can have a particle effect of its own, if you like. Feel free to modify the particle section of the script to suit your preferences. You need only include the word "Sparkle" in the Name field of the pad prim to activate the particle effect when the pad is on. The particle effect is automatically turned off when the pad is turned off. And, it will scale itself automatically based on the X value of the prim size.

It is also important to note that the pad prim you use for your object should always be oriented with its Z-axis pointing up and the prim itself should be no larger on the z-axis than the original pad. Use a transparent pad prim, if you need the base of your transporter to be taller than the original pad. It is okay to rotate the pad prim on the z-axis, but not the x or y axes.

Have you created your own custom beam particle effect that you think would make a good addition to the selection of beam styles already offered?

Contact me, and if I like it and think it would make a good addition to the system, I will pay you a fair negotiated price for the rights to include the beam style with future versions of the transporter system.

Have you created a really cool looking transporter pad, booth, vehicle, or other object that you would like to sell with my transporter system scripts in it?

Contact me, and if it meets the specs of the TP on/off script without modification, or does not need it at all, and I like it, we can sell it under the Stratotech name and I will pay you 35% of the revenue generated from the sale of your model! Sorry, I offer no "Developer Kit" or "Reseller Kit" for my transporter system. The only way I can maintain pricing integrity and ensure that all customers are always offered the most current release of the product is to maintain exclusive control over the distribution of the scripts.

If you're working on scripts or other related objects and need additional information about how the system works or how you might send or receive additional information to or from the system, let me know and I'll be glad to help in any way I can.