InSim Relay

From LFS Manual
Jump to navigationJump to search

This page contains outdated information, but is kept for archival reasons. Since 13th August 2025, relay service has been shut due to causing maintainence baggage, while having little to none use by the community. See: Proposed end of InSim relay (https://www.lfs.net/forum/thread/112079) forum thread.

InSim Relay usage information

The InSim Relay is a service that can connect to your LFS host via InSim and relay the InSim information sent by your host, to anyone who connects to the InSim Relay. This relayed data can be used by programmers for various things, such as the LFS Remote (remote viewing / adminning of a race) and race-tracking to store race information and statistics.

To have your host connected to the Relay, see this page on LFS World: http://www.lfsworld.net/?win=hosts&whichTab=insim_relay

The rest of this document is only for programmers who want to know how to connect to the InSim Relay, so they can make use of the available data.

Source file information: InSim Relay source

Connecting to the InSim Relay

The Relay code below can be seen as an extension to the regular InSim protocol, as the packets are constructed in the same manner as regular InSim packets and use their own identifiers.

Connect your client to isrelay.lfs.net:47474 with TCP. After you are connected you can request a list of hosts, so you can see which hosts you can connect to. Then you can send a packet to the Relay to select a host. After that the Relay will send you all insim data from that host.

Some hosts require a spectator password in order to be selectable.

You do not need to specify a spectator password if you use a valid administrator password.

If you connect with an administrator password, you can send just about every regular InSim packet there is available in LFS, just like as if you were connected to the host directly. For a full list, see end of document.

Packet types used for the Relay

```
#define IRP_ARQ         250     // Send : request if we are host admin (after connecting to a host)
#define IRP_ARP         251     // Receive : replies if you are admin (after connecting to a host)
#define IRP_HLR         252     // Send : To request a hostlist
#define IRP_HOS         253     // Receive : Hostlist info
#define IRP_SEL         254     // Send : To select a host
#define IRP_ERR         255     // Receive : An error number
```

To request a hostlist from the Relay, send this packet :

```
struct IR_HLR // HostList Request
{
        byte    Size;           // 4
        byte    Type;           // IRP_HLR
        byte    ReqI;
        byte    Sp0;
};
```

That will return (multiple) packets containing hostnames and some information about them

The following struct is a sub packet of the IR_HOS packet

```
struct HInfo // Sub packet for IR_HOS. Contains host information
{
        char    HName[32];      // Name of the host

        char    Track[6];       // Short track name
        byte    Flags;          // Info flags about the host - see NOTE 1) below
        byte    NumConns;       // Number of people on the host
};

// NOTE 1)
#define HOS_SPECPASS            1       // Host requires a spectator password
#define HOS_LICENSED            2       // Bit is set if host is licensed
#define HOS_S1                  4       // Bit is set if host is S1
#define HOS_S2                  8       // Bit is set if host is S2
#define HOS_FIRST               64      // info: http://www.lfsforum.net/showthread.php?p=1376118#post1376118
#define HOS_LAST                128     // info: http://www.lfsforum.net/showthread.php?p=1376118#post1376118
```

```
struct IR_HOS // Hostlist (hosts connected to the Relay)
{
        byte    Size;           // 4 + NumHosts * 40
        byte    Type;           // IRP_HOS
        byte    ReqI;           // As given in IR_HLR
        byte    NumHosts;       // Number of hosts described in this packet

        HInfo   Info[6];        // Host info for every host in the Relay. 1 to 6 of these in a IR_HOS
};
```

To select a host in the Relay, send this packet :

```
struct IR_SEL // Relay select - packet to select a host, so relay starts sending you data.
{
        byte    Size;           // 68
        byte    Type;           // IRP_SEL
        byte    ReqI;           // If non-zero Relay will reply with an IS_VER packet
        byte    Zero;           // 0

        char    HName[32];      // Hostname to receive data from - may be colourcode stripped
        char    Admin[16];      // Admin password (to gain admin access to host)
        char    Spec[16];       // Spectator password (if host requires it)

};
```

To request if we are an admin send:

```
struct IR_ARQ // Admin Request
{
    byte    Size;               // 4
    byte    Type;               // IRP_ARQ
    byte    ReqI;
    byte    Sp0;
};
```

Relay will reply to admin status request :

```
struct IR_ARP // Admin Response
 {
         byte   Size;           // 4
         byte   Type;           // IRP_ARP
         byte   ReqI;
         byte   Admin;          // 0- no admin; 1- admin
 };
```

If you specify a wrong value, like invalid packet / hostname / adminpass / specpass, the Relay returns an error packet :

```
struct IR_ERR
{
        byte    Size;           // 4
        byte    Type;           // IRP_ERR
        byte    ReqI;           // As given in RL_SEL, otherwise 0
        byte    ErrNo;          // Error number - see NOTE 2) below
};

// NOTE 2) Error numbers :
#define IR_ERR_PACKET1          1    // Invalid packet sent by client (wrong structure / length)
#define IR_ERR_PACKET2          2    // Invalid packet sent by client (packet was not allowed to be forwarded to
host)
#define IR_ERR_HOSTNAME         3    // Wrong hostname given by client
#define IR_ERR_ADMIN            4    // Wrong admin pass given by client
#define IR_ERR_SPEC             5    // Wrong spec pass given by client
#define IR_ERR_NOSPEC           6    // Spectator pass required, but none given
```

Regular insim packets that a relay client can send to host :

For anyone
TINY_VER
TINY_PING
TINY_SCP
TINY_SST
TINY_GTH
TINY_ISM
TINY_NCN
TINY_NPL
TINY_RES
TINY_REO
TINY_RST
TINY_AXI

Admin only
TINY_VTC
ISP_MST
ISP_MSX
ISP_MSL
ISP_MTC
ISP_SCH
ISP_BFN
ISP_BTN

The relay will also accept, but not forward
TINY_NONE // for relay-connection maintenance

Live for Speed guides and tutorials v

Basic Setup Guide | Advanced Setup Guide | Technical Reference | League Racing Guide | Skin Tutorial | Autocross Editor | LFS Editor Guides | Hosting | File Formats | LFS Programming | Movie Tutorial | Scripting | Translating

Retrieved from "https://en.lfsmanual.net/index.php?title=InSim_Relay&oldid=12241"

- This page was last edited on 5 September 2025, at 11:07.