

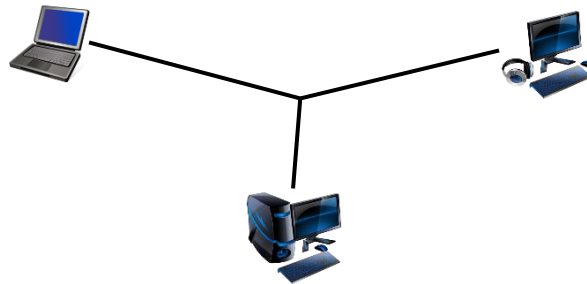


Lightrail Product Summary

Lightrail is an integration and communication framework for loosely coupled applications that delivers messages between systems. It builds on the latest operating system technology to provide top tier performance, a tiny memory footprint, and a zero-configuration deployment model*1.

Discovery

Applications using Lightrail do not need to be configured to talk to each other. There are no addresses, hosts, or servers to configure. Discovery utilizes UDP multicast to locate one another in real-time, then all communication occurs through a reliable TCP, binary protocol. This means systems can be added and removed in a running environment, the changes will be automatically recognized, and Lightrail will reconfigure itself without the user application ever knowing.



Connections between Systems is Dynamic, they are made as they are Present

Publish/Subscribe

Lightrail provides the ability to publish and subscribe messages, creating a one-to-many relationship, where any publisher can send a message which, in turn, can be received by any number of subscribers.

- Publishers and subscribers are linked with "Terminal" strings that are matched to one another, similar to a regular expression.
- Messages are sent directly from publisher to subscriber, there is no middleman. This provides enhanced security and performance.

*1: Applications in the same sub-network (subnet) require zero-configuration. Communication outside of a subnet is not yet supported, but is under development.

*2: There is a pseudo-random algorithm that provides selection which averages an even distribution over time.

*3: The C++ API supports binary, ASCII, UTF-8, and UTF-16 (Unicode) encodings. The .Net API supports all these plus UTF-32.



Request/Response

Lightrail also provides the ability for request/response messaging. There can be any number of listeners for a given request, where a single responder is selected for fulfilling a request*2.

- Requesters and responders are linked, like publish/subscribe, with "Terminal" strings.
- There is inherent load-balancing and fault tolerance when there are multiple responders for the same request.

Security

Lightrail provides message security with the Advanced Encryption Standard (AES) and 256-bit keys. This is provided on a message-by-message basis. A single application can send and receive messages with different encryptions, or no encryption, side-by-side. The AES is a standard recognized for use by the United States federal government and 256-bit encryption is secure enough to be used for Top Secret documents.

Messages

A message that is sent and received between applications is defined as any set of information. A message has no limitation on what it can contain. It can be a string, XML, or any binary data. Lightrail itself also provides several text encoding options for textual data*3.

Application Programming Interface (API)

There are currently two APIs for building against Lightrail.

- C++ class interface: A high-performance object-oriented API that has the feel of a modern language.
- .Net interface: An API for use with Microsoft .Net languages, such as C#, VB.Net, and C++.Net

Both APIs are simplistic enough to start sending and receiving messages in as few as 7 lines of code.

*1: Applications in the same sub-network (subnet) require zero-configuration. Communication outside of a subnet is not yet supported, but is under development.

*2: There is a pseudo-random algorithm that provides selection which averages an even distribution over time.

*3: The C++ API supports binary, ASCII, UTF-8, and UTF-16 (Unicode) encodings. The .Net API supports all these plus UTF-32.