

LIVE Server 5.0 Administrator's Guide

Table Of Contents

Welcome to LIVE!	1
LIVE at the Center	1
Some of the things you can do with LIVE sessions:	1
The LIVE Server Protocol.....	2
1. Getting started with LIVE	2
1.1. Some LIVE basics	2
1.2. Enjoy LIVE performance!.....	3
2. A deeper insight of the LIVE technology and the LIVE compression	4
2.1. The LIVE way of caching.....	4
2.2. Roundtrips suppression and bandwidth adaptive mechanism.....	5
3. The LIVE components at work	5
About LIVE Sessions.....	7
What is session Persistency	7
Session Mobility.....	7
Session Sharing	7
LIVE Console Sessions.....	8
Logging in to the LIVE Console.....	8
Reconnecting Remotely to a Console Session	9
LIVE Server Installation.....	11
Installation Options	11
Root Installations.....	11
Non Root Installations	11
Post-Installation Testing.....	11
Notes	11
Installation Instructions.....	13
LIVE Server AIX ppc Installation	13
RPM (root) Install File.....	13
Tar (non-root) Install File	13
System Requirements	13
LIVE Server HP-UX Installations.....	13
Depot (root) Install File	13
Tar (non-root) Install File	14
System Requirements	14
LIVE Server Linux x64 Installation	14
RPM (root) Install File.....	14
Tar (non-root) Install File	15

System Requirements	15
Linux x86 Installation	15
RPM (root) Install File.....	15
Tar (non-root) Install File	16
System Requirements	16
LIVE Server Solaris sparc Installation	16
Package (root) Install File.....	16
Tar (non-root) Install File	17
System Requirements	17
LIVE Server Solaris x86 Installation.....	17
Package (root) Install File.....	17
Tar (non-root) Install File	17
System Requirements	18
LIVE Server Configuration Options.....	19
File Format: LIVE Session.....	22
Required Tags	23
Contact StarNet.....	25
StarNet Main Office	25
Customer Support.....	25
Sales Department.....	25

Welcome to LIVE!

StarNet's LIVE series of X server solutions offers Windows and Linux users unprecedented cross-platform flexibility. You can display applications from remote Unix and Linux hosts securely and with persistency, thus vastly improving end-user productivity. For the first time Linux and Windows users can seamlessly and securely access their remote sessions from anywhere and from any workstation, whether it runs Windows or Linux. Working outside the LAN no longer comes at a big performance penalty as compression and SSH tunneling allow LIVE sessions to perform at LAN-like speed.

LIVE at the Center

Key to the LIVE functionality is StarNet's family of LIVE server packages for Linux, Solaris, AIX and HP-UX host computers. The LIVE server, a virtual proxy server, keeps sessions running while the remote Windows or Linux client is disconnected from the network – by accident or when the user suspends his session deliberately. The LIVE server also hosts the LIVE session configuration so that the end-user can connect to sessions from any workstation without having to remember session set-up settings.

Some of the things you can do with LIVE sessions:

1. On the Office Windows PC, X-Win32 connects to X applications on either the application host or on Linux PCs. The user can suspend those sessions and resume them on a Linux/Mac/Windows PC at home.
2. With LinuxLIVE, the Linux user suspends applications running on the local machine and on the remote host and resumes them on another Linux or Windows PC on the LAN or at home.
3. Linux, Macs and Windows PCs can share the same session.
4. Using LIVE Console, the Linux user can redisplay the console desktop on a Windows machine at home.
5. The Mac at home can connect to the LIVE server on a secure gateway server from which an SSH connection is made to applications on the host or Linux PC.
6. The Windows user at home can remotely start an application hosted on his Linux PC in the Office, suspend it and resume that session with LinuxLIVE on the Linux machine at work.

The LIVE Server Protocol

1. Getting started with LIVE

LIVE allows you to display X11 sessions running on remote Linux and Unix hosts on your PC, Mac, or Linux workstation even across slow or low-bandwidth network connections. This impressive performance versus traditional X connections (SSH, REXEC, etc.) comes from a combination of X protocol compression techniques and an integrated set of proxy agents that reduce X11 round-trips.



Fig 1. A remote X desktop displayed with LIVE on a local Windows Vista desktop

1.1. Some LIVE basics

The X protocol is at the base of communication between an X server and a graphical X client application on the Unix/Linux host. The X server is a program that is able to draw the graphical interface of the X client. The X server also handles keyboard and mouse events issued by the user and sends them back to the X client program, which then executes those commands.

When the X client needs to draw something on the screen, it issues a number of requests to the X server. About 160 different types of X requests, including extensions, are specified in the X protocol. Each request represents, for example, a primitive graphic element.

Some requests sent by the X client need a reply from the X server. Each request made by the X client, and the correspondent reply from the X server, constitutes a roundtrip. Roundtrips slow down the responsiveness of a graphical program because of the time needed by the requests to complete the two-way trip. As you get further and further

away from the X host the latency factor also increases and slows down the display of the X client on the PC even more.

The X protocol is network transparent. It doesn't care about where the X client and the X server physically reside. They may be on the same or different host machines. A local X server may draw on the local user's screen the graphical interface of a X client running remotely and can also send local mouse and keyboards events to the remote X client.

If you have a system account on a remote Linux/Unix machine, you can verify this by running, from your local xterm:

```
ssh -X remote_username@remote_hostname xterm
```

This establishes a channel for tunneling the X11 connection from the xterm application to the local X server of the user running the command. However, the overall effect is very similar to a direct connection for the scope of this example. A new xterm should appear on your screen, if the remote SSH is configured to allow for X forwarding, prompting the remote username and host you have connected to.

Please note that when a remote application is launched and its first window is displayed on the local screen, the number of roundtrips may be near to many thousands, giving a feeling of slowness and bad responsiveness. When the X client and the X server are running on the same host, the communication happens through UNIX domain sockets and roundtrips are reasonably fast. When, instead, the X client and the X server are running on different host machines, the interprocess communication is transported through TCP/IP network sockets and the network. In this case, roundtrips cause a slowness that is much heavier than in the previous case.

An additional issue that can affect performance is the network latency. The quality of each link type is made up of 2 parameters: network bandwidth and its latency.

Bandwidth refers to the rate of bytes that can be transferred per second while the network's latency describes how much time each packet of data needs to travel from one end to the other. Typically, a modem link has a latency of 200 to 500 milliseconds. An ADSL link has a latency of about 50 milliseconds. A local Ethernet LAN link's latency is less than 1 millisecond. A UNIX domain socket link is well below 0.1 milliseconds. You can test the latency of any network link with the ping command. The ping command shows the round-trip time in your terminal window.

1.2. Enjoy LIVE performance!

Using the LIVE method users get the best performance when the X client and the X server are running on different host machines because LIVE offers:

- An efficient compression of the X traffic.
- The cache mechanism to store and re-use data transferred between server and client
- A drastic reduction of time-consuming X roundtrips, bringing their total number close to zero.

If you want to see the difference LIVE makes in applying these techniques, you can play with various configurations via the LIVE Client GUI. For example, if you connect over a DLS Internet connection and you select a Connection Speed "LAN" in the LIVE Client GUI -> General tab, neither the LIVE Compression and the lazy encoding techniques will be used. Caching and round trip suppression are used instead. Performance can be a bit sluggish.

Next, select link type ADSL to see all these techniques applied and performance increases significantly, be it at a slight loss of graphics quality do to the data compression

2. A deeper insight of the LIVE technology and the LIVE compression

A normal X session, browsing the Internet or accessing common desktop applications, generates hundreds of megabytes of protocol data. Efficient compression is not only needed to run applications over slow-bandwidth links, but also to run multiple user sessions on common corporate LANs.

LIVE's goal is to permit users to run, over the Internet, the same graphic intensive applications that run on their office desktop computers. We don't try to pose any requirement on which X applications can or cannot be executed. To effectively use other remote display solutions, users have to disable fancy backgrounds, drop-down menus' animations and similar graphics effects. LIVE was designed to deal with such "extreme" conditions without users or desktop application developers having to modify their habits or their code. This has been our most important requirement since the beginning.

Everybody who has worked on enabling X-Window protocol to operate over low-bandwidth, high-latency links will say that the problem doesn't only reside in compression. In fact, LIVE X compression is just one part of LIVE proxy. An important part of LIVE design and implementation was dedicated to reducing round-trips and implementing strict flow-control of data traveling through the low-bandwidth proxy link.

LIVE X protocol compression is derived from DXPC. The DXPC - Differential X Protocol Compressor project, released in 1995 by Brian Pane, was not only an invaluable source of ideas and a very good basis on which LIVE X compression was founded, it also offered specific differential encoding of many of the nearly 160 requests, replies and events that constitute the core X protocol.

2.1. The LIVE way of caching

Alongside the LIVE compression, there is the LIVE innovative way of caching – LIVE splits the X message into two parts, an identity and a data part. LIVE maintains a cache of the last X messages sent through the wire in the main memory, divided by protocol opcode. This cache is named MessageStore.

To allow a fast look-up of messages in the specific MessageStore, LIVE calculates a checksum of any new request or reply that has to be encoded. Any message type has its own method to calculate the checksum of the identity, while the checksum of the data part is simply obtained by adding any data byte to the checksum. The MessageStore's method of calculating checksum of identity has to be carefully chosen to not include those fields that are likely to change across different instances of the same X request.

When a new X request is received, LIVE calculates the checksum of the new message and searches it in the MessageStore. If the message is found, LIVE only sends this status information to the remote peer, together with the position where the message can be retrieved from store and a differential encoding of all those fields that are not part of the identity checksum. The idea of sending X updates using per-message differential algorithms were already present in the old DXPC. But the idea of splitting X messages

into a 'fingerprint' and a 'data' part and caching the data part as a whole is completely new. This is what achieves the great compression boost of LIVE.

Over the years, methods have been fine-tuned by applying a modified encoding to each of the nearly 160 X message opcodes. Now LIVE is in the range of 60% to 80% cache hits for the overall sample of X messages that go across the wire. For some messages, like graphic requests, images, fonts, icons and other requests used in common office automation desktop applications, cache hits can reach 100%, allowing LIVE to achieve effective compression ratios in the order of 1000:1.

2.2. Roundtrips suppression and bandwidth adaptive mechanism

Think of running Mozilla 1.1 through plain X. This can require several minutes of waiting because Mozilla requires almost 6000 roundtrips to start up. With LIVE, start-up time decreases to less than 20 seconds, thanks to the round-trip suppression that reduces roundtrips to about one dozen. Even on slow or low-bandwidth network connections, you can still get impressive performance thanks to LIVE's lazy encoding algorithm and LIVE's capability to automatically tune itself to network bandwidth and latency parameters.

3. The LIVE components at work

LIVE is made up of a number of components that can be easily installed thanks to the packages that StarNet offers for a great number of platforms. The communication between the local and the remote computer is achieved by setting a two-way proxying system.

Both proxies, provided by the installation of StarNet's LIVE clients and the LIVE Server component on the remote host, communicate using the LIVE protocol. This is based on X11, but with the added extensions for compression and caching in order to reduce the round-trip X traffic between the LIVE proxies to near-zero.

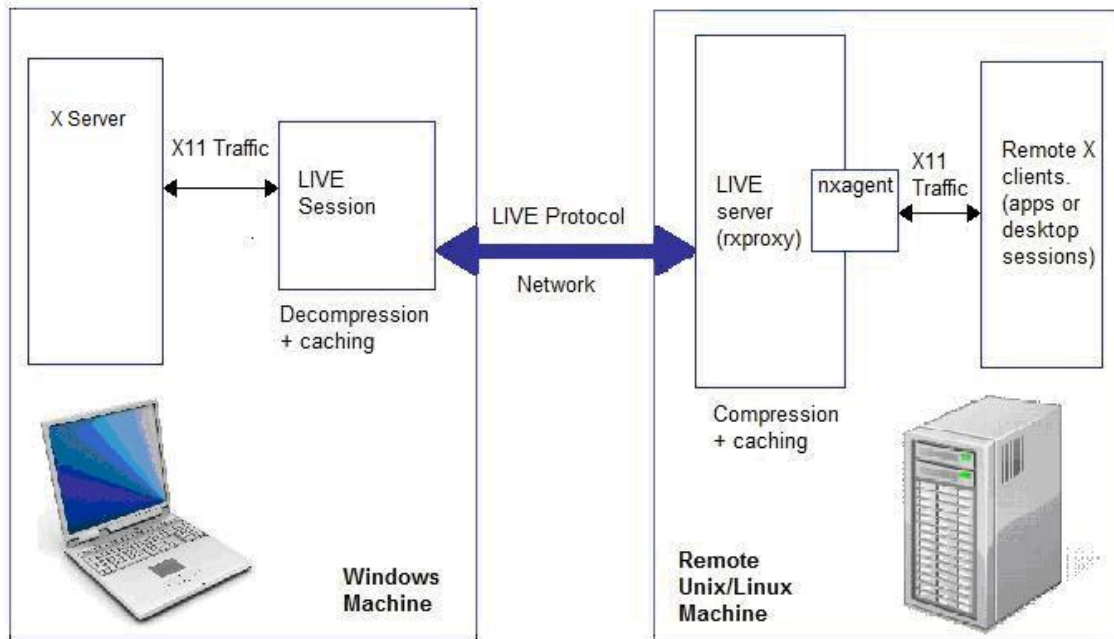


Fig 2. The LIVE proxying system.

The local proxy (LIVE Client) talks with the X Server and is in charge of translating the LIVE protocol back to X11. The remote proxy (the LIVE Server) talks to the remote X11 applications and uses the X11 protocol. Also on the remote side, no modifications are needed to the X server.

The remote LIVE proxy poses itself to the X clients as if it is the X server. All the roundtrips take place at this point and since they happen on the same host machine, they are quickly resolved through the UNIX domain sockets. The remote proxy, in fact, incorporates a kind of X server, named **nxagent**, that becomes the X server of all the remote X clients. The **nxagent**, also named as the **X11 agent**, translates the X11 protocol into LIVE protocol, receiving X11 requests as drawing commands and sending them to the local proxy. At this stage, the LIVE protocol, used for carrying compressed data or references to cached elements through the network, doesn't contain roundtrips.

Both the local and remote proxy keep their own cache of transferred data. These two sets of cache are synchronized to be useful for saving all further transmission of pixmaps, icons and so on between the remote and the local end points.

About LIVE Sessions

LIVE sessions are special sessions that allow [session persistency](#). These sessions can be suspended and resumed at a later time or terminated ending the session. The LIVE Server runs on the remote host allowing the session to stay active even after the LIVE client has disconnected.

The LIVE server starts automatically when a LIVE client connects. The LIVE client connects to the LIVE server via SSH. The client runs the rxsci5 program, which in turn starts rxserver-bin, if it isn't already running. Communication between the LIVE components on the server are through files and named sockets in a directory named .nx-{-hostname} in your home directory, which is created automatically. You do not need to start anything after installation. It will automatically exit when you terminate your last running session.

For security reasons, each user runs his own instance of the LIVE server (rxserver-bin). The LIVE Server is available for Linux, Linux-64, Solaris, Solaris-x86, HPUX, and AIX.

LIVE sessions require the LIVE server component to be installed on the remote UNIX/Linux host in order to function

What is session Persistency

Session persistency means that even though X-Win32 has disconnected from the host machine, the LIVE session is still actively running on the host machine.

For example, you can start a long running compilation project through a LIVE session, suspend the session, and reconnect at a later time when the compilation has finished. With other connection methods closing the window will automatically terminate your session requiring you to start the compilation over from the beginning.

Session Mobility

Sessions can be transferred from computer to computer allowing session mobility. A session can also be started on a Unix/Linux machine and reconnected to from a windows machine. For complete mobility, sessions can be stored on the remote Unix/Linux machine.

Session Sharing

Multiple users can now connect to the same LIVE session to work and collaborate together at the same time. Users can choose to take control of the desktop, or they can simply view the desktop itself.

LIVE Console Sessions

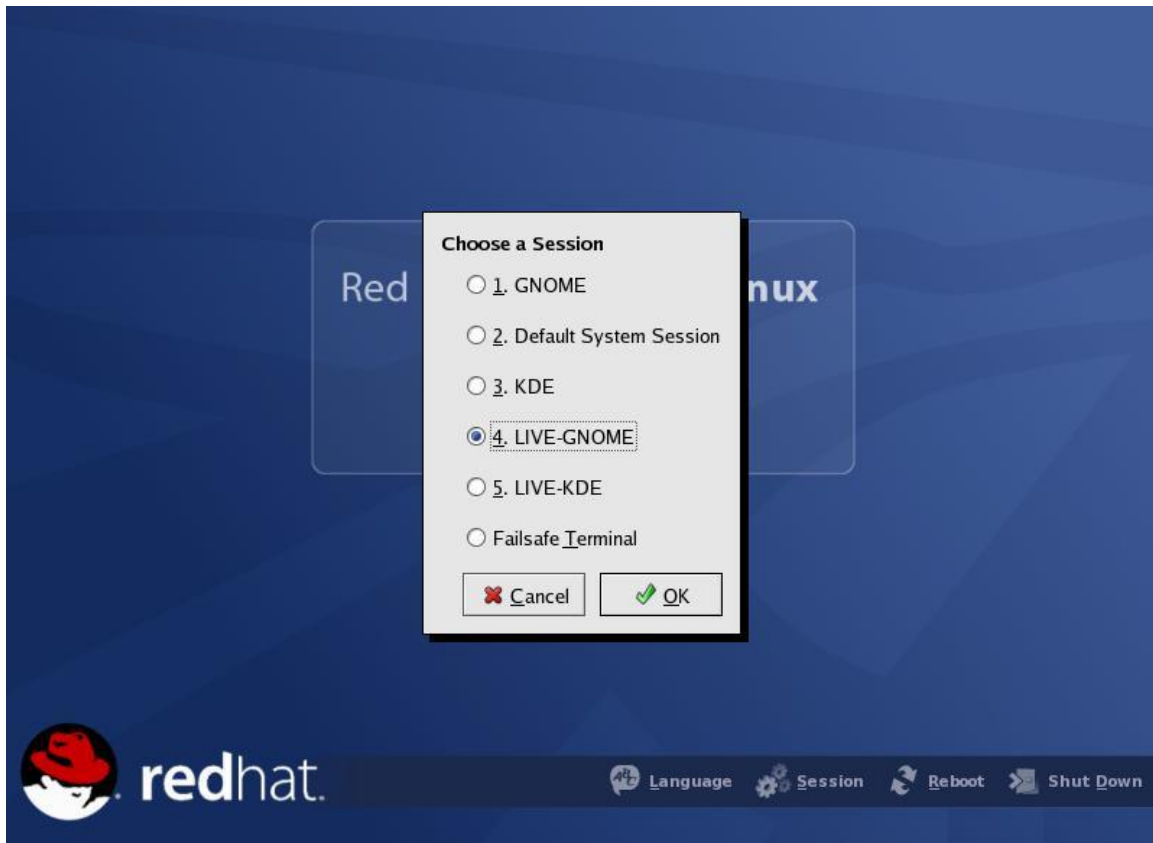
LIVE Console is the first X server solution to allow Linux and Sun Solaris workstation desktops to be re-displayed on Windows PCs, Macs, or other Linux workstations. No other X server can do this. LIVE Console is the first in a series of connectivity tools for Linux and Unix-based workstation users. Besides the redisplay of their desktop LIVE Console offers two additional key functionalities:

- Suspend/Resume Desktop – For security reasons, many companies require desktops be logged out when the user leaves his office for meetings or goes home. That means he has to terminate any running applications. LIVE Console allows the user to log-out of his Linux/Solaris desktop and log back in at a later time while open applications continue to run in the background.
- Simultaneous Multiple User Accounts – With LIVE Console, multiple consoles can be running simultaneously. One user can log out to allow another user to work on a workstation. The first user's desktop and the applications on it continue to run in the background.

Installing the LIVE Server as root automatically grants end user access to the LIVE Console. The desktop session runs in full screen mode just like a normal session. However, since it is running through the LIVE server, the session is persistent. Users can Suspend their sessions (by pressing CNTL+ALT+T) allowing others to log in to the same physical computer. User can also reconnect to their sessions remotely using any of the LIVE clients.

Logging in to the LIVE Console

When logging in directly from the Unix/Linux side, click on the option to change your session. New session types will be placed in the Session Chooser prefaced with LIVE (for example LIVE-Gnome). Select the LIVE session and log in normally.



Users can also start a Console Session from a remote host. See the next section for details.

Reconnecting Remotely to a Console Session

1. Create a LIVE Connection.
2. Connect to the remote host.

3. In the Instance Browser (top Window), double click on "Console-{desktop type}"

The screenshot displays the Instance Browser interface. At the top, there is a table titled "Instances" with the following data:

Session	Status	Started	Last Suspended	Last Resumed	Window Size
Terminal	Working...	7/9/2009 4:39:4...	7/9/2009 6:23:1...	7/9/2009 4:41:2...	
Shared	Suspended	5:36:09 AM	5:37:30 AM		800x600
KDE Session	Running	5:39:35 AM			1024x768

Below the table is a "Shadow" button. Underneath is a section titled "Sessions" containing four desktop icons: "GNOME", "KDE", "GNOME", and "CUSTOM". Below these icons are labels: "Console-...", "KDE Session", "Shared", and "Terminal". At the bottom of the interface are buttons for "Launch", "Add", "Edit", and "Remove".

4. You will be reconnected to the running session.

LIVE Server Installation

Installation Options

The LIVE Server is available at the StarNet website:

LIVE Server 5.0: <http://www.starnet.com/support/live/server.php?version=5.0>

For step by step instructions to install the LIVE Server, please refer to the System Specific Installation Instructions. The LIVE Server is available for [Linux](#), [Linux-64](#), [Solaris](#), [Solaris-x86](#), [HPUX](#), and [AIX](#).

Root Installations

If you are an administrator on the Linux machine on which you will install the Live Server, it is recommended that you use the root installation. This will install using RPM (or other system specific packages), which will automatically check dependencies, and install the server into the place where all users can test it.

Non Root Installations

Individual users who do not have root access can still test the LIVE functionality by installing the non-root package, which is a compressed tar file that will be installed in your home directory. The *Remote Server Interface Program* of the LIVE Connection must be edited in order to point to the the location of the LIVE Server Installation (rxlaunch sci5)

Post-Installation Testing

Ensure that sshd is installed on your remote system. You can test this out by running the command: ssh localhost. If you can log in through localhost your system has ssh installed

To test if the LIVE Server was installed correctly, run the command as a **non-root user**: rxlaunch sci5.

A message such as

```
<?xml version="1.0" encoding="UTF-8"?><Messages
```

should appear.

If no errors appear, the LIVE Server is running correctly. Otherwise, check to see that you have met the system requirements for your Unix OS and have all the proper libraries installed.

Notes

The LIVE server starts automatically when a LIVE client connects. The LIVE client connects to the LIVE server via SSH. The client runs the rxsci5 program, which in turn starts rxserver-bin, if it isn't already running. Communication between the LIVE components on the server are through files and named sockets in a directory named .nx-

LIVE Server 5.0 Administrator's Guide

{-hostname} in your home directory, which is created automatically. You do not need to start anything after installation. It will automatically exit when you terminate your last running session.

For security reasons, each user runs his own instance of the LIVE server (rxserver-bin).

Installation Instructions

LIVE Server AIX ppc Installation

RPM (root) Install File

- Is a RPM file.
- Will install in /usr/bin and /usr/lib, or wherever you have the system BIN and LIB directories on your system.
- Requires root (system) access to install.
- Install by running (as root): rpm -i LIVEServer-###.aix5.3.ppc.rpm
- ### is the build number

Tar (non-root) Install File

- Is a gzip'ed tar file.
- Will typically install in your home directory, under a directory titled 'LIVEServer'.
- This path needs to be added to the LIVE session on the client.
 1. Edit your LIVE session in X-Win32
 2. Click on the Advanced Tab
 3. Change Remote Server Interface Protocol from: **rxlaunch sci5** to **LIVEServer/bin/rxlaunch sci5** (or the location of rxlaunch)
- Does not require root (system) access to install or run.
- Install by running (in your home directory): gunzip -dc LIVEServer-###.aix5.3.ppc.tar.gz | tar xf
- ### is the build number

System Requirements

- CPU: IBM Power PC
- OS: IBM AIX 5 or greater. Currently tested on IBM AIX 5.3.
- libjpeg, libpng, libz packages available on the "AIX Toolbox for Linux Applications for POWER Systems" installation CD.
- sshd must be enabled on the remote system
- Enable TCP forwarding in /opt/ssh/etc/sshd_config by changing the "AllowTcpForwarding" line from "no" to "yes": AllowTcpForwarding yes

LIVE Server HP-UX Installations

Depot (root) Install File

- Is a gzip'd tar file of an HP-UX depot directory.
- Download LIVE HP-UX Server
- Requires root (system) access to install.
- Install by running (as root):
- `tar xzf live-###-hpux.tar.gz`
- `swinstall -s `pwd`/live-###/`
- `###` is the build number

Tar (non-root) Install File

- Is a gzip'ed tar file.
- Download LIVE HP-UX Server (non-root install)
- Will typically install in your home directory, under a directory titled 'LIVEServer'.
- This path needs to be added to the LIVE session on the client.
 1. Edit your LIVE session in X-Win32
 2. Click on the Advanced Tab
 3. Change Remote Server Interface Protocol from: **rxsci** to **LIVEServer/bin/rxsci** (or the location of rxsci)
- Does not require root (system) access to install or run.
- Install by running (in your home directory): `tar xzf live-###-hpux-nonroot.tar.gz`

System Requirements

- CPU: PA-RISC
- OS: HP-UX 11.11.
- GCC 4.2 or later (libstdc++.sl.6). Get it [here](#)
- sshd must be enabled on the remote system
- Enable TCP forwarding in sshd_config by changing the "AllowTcpForwarding" line from "no" to "yes": `AllowTcpForwarding yes`

LIVE Server Linux x64 Installation

RPM (root) Install File

- Is a RPM file.
- Will install in `/usr/bin` and `/usr/lib`, or wherever you have the system BIN and LIB directories on your system.
- Requires root (system) access to install.
- Will automatically check for system dependencies.

- Install by running (as root): `rpm -i LIVEServer-###.x86_64.rpm`
- `###` is the build number

Tar (non-root) Install File

- Is a gzip'ed tar file.
- Will typically install in your home directory, under a directory titled 'LIVEServer'.
- This path needs to be added to the LIVE session on the client.
 1. Edit your LIVE session in X-Win32
 2. Click on the Advanced Tab
 3. Change Remote Server Interface Protocol from: **rxlaunch sci5** to **LIVEServer/bin/rxlaunch sci5** (or the location of rxlaunch)
- Does not require root (system) access to install or run.
- System dependencies must be checked manually (see below).
- Install by running (in your home directory): `tar xzf LIVEServer-###.x86_64.tar.gz`
- `###` is the build number
- On RedHat Linux, make sure `/usr/X11R6/bin` (where `xauth` is found) is in your path. This may involve editing your `.bashrc` file or, as an alternative, the `rxsci` script.

System Requirements

- CPU: Intel x86_64 or AMD64
- OS: Linux. Currently tested on Debian 64, RHEL4-64, RHEL5-64
- `sshd` must be enabled on the remote system
- Enable TCP forwarding in `/etc/sshd/sshd_config` by changing the "AllowTcpForwarding" line from "no" to "yes": `AllowTcpForwarding yes`

Linux x86 Installation

RPM (root) Install File

- Is a RPM file.
- Will install in `/usr/bin` and `/usr/lib`, or wherever you have the system BIN and LIB directories on your system.
- Requires root (system) access to install.
- Will automatically check for system dependencies.
- Install by running (as root): `rpm -i LIVEServer-###.i386.rpm`
- `###` is the build number

Tar (non-root) Install File

- Is a gzip'ed tar file.
- Will typically install in your home directory, under a directory titled 'LIVEServer'.
- This path needs to be added to the LIVE session on the client.
 1. Edit your LIVE session in X-Win32
 2. Click on the Advanced Tab
 3. Change Remote Server Interface Protocol from: **rxlaunch sci5** to **LIVEServer/bin/rxlaunch sci5** (or the location of rxlaunch)
- Does not require root (system) access to install or run.
- System dependencies must be checked manually (see below).
- Install by running (in your home directory): `tar xzf LIVEServer-###.i386.tar.gz`
- ### is the build number
- On RedHat Linux, make sure `/usr/X11R6/bin` (where `xauth` is found) is in your path. This may involve editing your `.bashrc` file or, as an alternative, the `rxsci` script.

System Requirements

- CPU: Intel x86 32 bit
- OS: Linux. Currently tested on Debian, RHEL4, RHEL5, SuSE 9, SuSE10
- `sshd` must be enabled on the remote system
- Enable TCP forwarding in `/etc/sshd/sshd_config` by changing the "AllowTcpForwarding" line from "no" to "yes": `AllowTcpForwarding yes`

LIVE Server Solaris sparc Installation

Package (root) Install File

- Is a Solaris Package file.
- Requires root (system) access to install.
- Install by running (as root): `gunzip live-###-sol8-sparc-local.gz && pkgadd -d live-###-sol8-sparc-local`
- Download extra libraries from <http://www.starnet.com/support/live/server.php#solaris>
- Extract the extra libraries into `/usr/lib/RX/lib` by running (as root): `gunzip -dc libs-sol10-sparc.tar.gz | tar xf - -C /usr/lib/RX/lib`
- Enable TCP forwarding in `/etc/sshd/sshd_config` by changing the "AllowTcpForwarding" line from "no" to "yes": `AllowTcpForwarding yes`
- ### is the build number

Tar (non-root) Install File

- Is a gzip'ed tar file.
- Will typically install in your home directory, under a directory titled 'LIVEServer'.
- This path needs to be added to the LIVE session on the client.
 1. Edit your LIVE session in X-Win32
 2. Click on the Advanced Tab
 3. Change Remote Server Interface Protocol from: **rxlaunch sci5** to **LIVEServer/bin/rxlaunch sci5** (or the location of rxlaunch)
- Does not require root (system) access to install or run.
- Install by running (in your home directory): `gunzip -dc live-###-sol8-sparc-local.tar.gz | tar xf -`
- **###** is the build number

System Requirements

- CPU: Sparc 64-bit
- OS: Solaris 8 or greater. Currently tested on Solaris 8 and Solaris 10.
- libiconv (libiconv.so.2). Get it from <http://www.starnet.com/support/live/server.php#solaris>
- GCC 3.4 or later (libstdc++.so.6 and libgcc_s.so.1).
- sshd must be enabled on the remote system
- Enable TCP forwarding in /etc/sshd/sshd_config by changing the "AllowTcpForwarding" line from "no" to "yes": AllowTcpForwarding yes

LIVE Server Solaris x86 Installation

Package (root) Install File

- Is a Solaris Package file for the Intel x86 platform.
- Will install in /usr/bin and /usr/lib, or wherever you have the system BIN and LIB directories on your system.
- Requires root (system) access to install.
- Install by running (as root): `gunzip live-###-sol10-i386-local.gz && pkgadd -d live-###-sol10-i386-local`
- **###** is the build number

Tar (non-root) Install File

- Is a gzip'ed tar file.
- Will typically install in your home directory, under a directory titled 'LIVEServer'.

- This path needs to be added to the LIVE session on the client.
 1. Edit your LIVE session in X-Win32
 2. Click on the Advanced Tab
 3. Change Remote Server Interface Protocol from: **rxlaunch sci5** to **LIVEServer/bin/rxlaunch sci5** (or the location of rxlaunch)
- Does not require root (system) access to install or run.
- Install by running (in your home directory): `gunzip -dc live-###-sol10-i386-local.tar.gz | tar xf -`
- **###** is the build number

System Requirements

- CPU: Intel Pentium
- OS: Solaris 10.
- sshd must be enabled on the remote system
- Enable TCP forwarding in `/etc/sshd/sshd_config` by changing the "AllowTcpForwarding" line from "no" to "yes": `AllowTcpForwarding yes`

LIVE Server Configuration Options

LIVE Server configuration options can be overridden by creating a file named LIVE.conf

For user specific overrides: **\$HOME/.config/StarNet/LIVE.conf**

For system wide overrides: **/etc/xdg/StarNet/LIVE.conf**

The format for an option is as follows: **{Option}={Value}**

The following options are available

Note: options not specified in the LIVE.conf file will take their default settings

Property	Description
ServerDirectory	<p>DEFAULT: {user's home directory}/.nx-@HOSTNAME@</p> <p>There are several variables that this option can interpret:</p> <p>@USER@ — variable which will be replaced with the username</p> <p>@HOSTNAME@ — variable which will be replaced with the hostname</p> <p>This directory contains the following files:</p> <p>rxsci-XXXX.log – temporary rxsci log</p> <p>rxserver.log – LIVE Server log file</p> <p>C-{UID} – directory containing running LIVE session information</p> <p>S-{UID} – directory containing running LIVE Console information</p> <p>cache-{session type} – directory containing the cache of each specific session</p>
SystemSessionsDirectory	<p>DEFAULT: /etc/xdg/StarNet/sessions/</p> <p>Specifies the directory on the host where global server-side LIVE sessions will be stored.</p>
UserSessionsDirectory	<p>DEFAULT: {user's home directory}/.nx-@HOSTNAME@/sessions/</p> <p>Specifies the directory on the host where user server-side LIVE sessions will be stored.</p> <p>There are several variables that this option can interpret:</p> <p>@USER@ — variable which will be replaced with the username</p> <p>@HOSTNAME@ — variable which will be replaced with the hostname</p>
LogLevel	<p>DEFAULT: 3</p> <p>0=fatal, 1=critical, 2=warning, 3=debug</p> <p>Specifies the rxserver.log and rxsci.log ogging level.</p>
KeepLogs	<p>DEFAULT: false</p> <p>Do not erase the rxsci-XXXX.log file even after the LIVE Client has disconnected successfully. Change to true for</p>

	session debugging
KeepSessions	DEFAULT: false Keeps the C-{session UID} and S-{session UID} directories even after the sessions have terminated successfully. Change to true for session debugging
IdleTimeout	DEFAULT: 60 Specifies the length of time in seconds the LIVE Server (rxserver-bin) will run before it is shut down when there are no more LIVE sessions
DisplayBase	DEFAULT: 1000 Specifies the base X DISPLAY number for the LIVE server. Change this option if there is already a service listening on this port.
DisplayLimit	DEFAULT: 99 Specifies the number of running LIVE Sessions the rxserver-bin process can handle.
SuspendTimeout	DEFAULT: -1 (no limit) Specifies the length of time in seconds a session can stay suspended until it is terminated by the server.
GNOMEPath	DEFAULT: gnome-session Specifies the default path of the GNOME startup script. Change to the fully qualified path of gnome-session if the startup script is not in the default \$PATH. Gnome typically installs in /usr/bin/gnome-session
KDEPath	DEFAULT: startkde Specifies the default path of the KDE startup script. Change to the fully qualified path of gnome-session if the startup script is not in the default \$PATH. For example SuSE users may want to set this option to /opt/kde3/bin/startkde when installing the system. This allows all clients to simply select KDE rather than qualifying the startkde path
CDEPath	DEFAULT: Xsession Specifies the default path of the CDE startup script. Change to the fully qualified path of gnome-session if the startup script is not in the default \$PATH. CDE typically installs in /usr/dt/bin/Xsession
TempDirectory	DEFAULT: /tmp/live-@USER@ Specifies where the lock file and the rxlisten socket file will be held. Do not change this option unless you are having issues reconnecting to a network mounted filesystem. There are several variables this option can interpret: @USER@ — variable which will be replaced with the username

	@HOSTNAME@ — variable which will be replaced with the hostname
--	---

File Format: LIVE Session

Sessions stored on the host can be user specific or global to all users. The directories are specified in LIVE.conf (see [LIVE Server Configuration Options](#)) Each file is named with a global unique identifier (guid).

The file is an XML based file with the following format:

```
<Session>
```

```
  <Uid>{global unique identifier}</Uid>
```

```
  <Name>Session Name</Name>
```

```
  <AppChoice>[GNOME|KDE|CDE|OTHER]</AppChoice>
```

```
  <AppPath>[/path/to/application]</AppPath>
```

```
  <Share>[true|false]</Share>
```

```
  <WindowMode>[Regular|Fullscreen|Rootless]</WindowMode>
```

```
  <WindowSize>
```

```
    <Width>width in pixels</Width>
```

```
    <Height>height in pixels</Height>
```

```
</WindowSize>
```

```
<WindowDepth>[8|16|24|32]</WindowDepth>
```

```
<UseRender>[true|false]</UseRender>
```

```
<DPI>[75|96|120]</DPI>
```

```
</Session>
```

Required Tags

The following options are required when creating a new Session

Property	Description						
<Uid>	{string} A global unique identifier GUID. Example {41f29734-3e65-475f-8700-f375ef6ed3af}						
<Name>	{string} The session name that appears in the LIVE server session browser.						
<AppChoice>	[GNOME KDE CDE OTHER] The desktop that the user selected						
<AppPath> (optional)	{string} The path to the application if OTHER desktop session type was selected. This is the field to the right of the Desktop in the session browser						
<Share>	[true false] Allow Desktop sharing. Multiple users can connect to the same session and view the same desktop. DEFAULT: false						
<WindowMode>	[Fullscreen Regular Rootless] Specifies the Window mode to use. DEFAULT: Fullscreen <table border="0"> <tr> <td>Fullscreen</td> <td>takes up the entire screen as if you were at the Console</td> </tr> <tr> <td>Regular</td> <td>(Default in session browser) opens up the session in a window</td> </tr> <tr> <td>Rootless</td> <td>Multiple Window Mode. Use this option when running single applications in Custom</td> </tr> </table>	Fullscreen	takes up the entire screen as if you were at the Console	Regular	(Default in session browser) opens up the session in a window	Rootless	Multiple Window Mode. Use this option when running single applications in Custom
Fullscreen	takes up the entire screen as if you were at the Console						
Regular	(Default in session browser) opens up the session in a window						
Rootless	Multiple Window Mode. Use this option when running single applications in Custom						

	sessions
<Width>	{int} Specifies the Desktop/Window width in pixels. DEFAULT: 1024
<Height>	{int} Specifies the Desktop/Window height in pixels DEFAULT: 768
<UseRender>	[true false] Turn on Render extension. Recommended to be left as true. DEFAULT: true
<DPI>	[75 96 120] Specify the DPI settings of the LIVE server. If fonts appear small, change the DPI to a higher value. DEFAULT: 75

Contact StarNet

StarNet Main Office

Web Site

<http://www.starnet.com>

Address

StarNet Communications
1270 Oakmead Parkway, Suite 301
Sunnyvale, CA 94085-4044

Customer Support

Web Site

<http://www.starnet.com/support>

- FAQs, latest upgrades, and new information regarding X-Win32's most recent developments.
- A customer support ticket system you can access online through X-Win32.

Sales Department

Telephone

(408) 739 0881

The Sales Department is available 8:00 AM - 5:00 PM PST, Monday through Friday.

Fax

(408) 739 0936

Email

SalesDept@StarNet.com