Exceed TurboX

Installation and Configuration Guide

V12.5.3

Table of Contents

1. Installation and Configuration Guide

Rocket® Exceed TurboX (ETX) is a web-based platform that enables users to launch UNIX and Windows desktops and applications over the internet. Exceed TurboX provides IT with a central platform to monitor and manage user access to systems. It provides users with a seamless experience, where desktops and applications running on remote hosts look and feel like they are running locally.

This document describes how to install and configure Exceed TurboX for initial use. It also provides information about how to upgrade to the latest version of the product. This document is intended for Exceed TurboX administrators.

2. Getting started with Exceed TurboX

Getting started with Exceed TurboX

This section gives an overview of the installation of the various Exceed TurboX components.

This section contains the following topics:

About server-side installation of Exceed TurboX

This section lists the server-side installation and configuration tasks you need to perform to set up Exceed TurboX. Complete the tasks in the following order:

1. Installing Exceed TurboX Server
2. Setting up the Exceed TurboX site
3. Installing Exceed TurboX Connection Nodes
4. Configuring Exceed TurboX for initial use

For descriptions of Exceed TurboX components, see About Exceed TurboX server-side components.

About Exceed TurboX server-side components

About Exceed TurboX server-side components

This section describes the server-side components that you need to install. Before installing and managing Exceed TurboX, familiarize yourself with these components:

- Exceed TurboX Connection Nodes
- Exceed TurboX datastore
- Exceed TurboX Server

Exceed TurboX Connection Nodes

Nodes are the processing hubs for the product, handling client requests to start sessions and starting a proxy with a unique display ID for each new request to launch a session. Depending on the role you assign to a node, it can also act as a user authenticator, proxy manager, or both.

Exceed TurboX infrastructure includes at least one node. If user load is high, you can install multiple Exceed TurboX Connection Nodes. Installing multiple Exceed TurboX Connection Nodes has the following benefits:

- **Load balancing**: In multi-node environments, you can have multiple authenticators and proxy managers. This allows you to utilize all resources by distributing the load among nodes, based on the criteria (defined by the administrator) that best suits the needs of the environment. New session requests are automatically distributed to the nodes that have the most available resources, in order to maximize stability and performance. For more information about load balancing options, see "Configuring general site settings" in the *Exceed TurboX Server Manager Help*.
- **Ease of maintenance**: If a node requires maintenance, an administrator can disable it temporarily during the maintenance period, and session requests will be forwarded to other nodes in the multi-node environment.

You can use custom load balancing scripts to control the load on nodes based on their own custom load balancing criteria, or to select a target node dynamically when launching a session, based on the profile's target node selection criteria. For details, see "Configuring load balancing" in the *Exceed TurboX Server Manager Help*.

Exceed TurboX datastore

Exceed TurboX Server uses a datastore to save all the configuration settings and files required to run the site.

If the datastore is not already setup (for example, if the datastore is restored from a backup) when Exceed TurboX Server is started, a new datastore will be initialized automatically, with standard settings, including some default profiles and templates. When an Exceed TurboX Server cluster is created with the Exceed TurboX Server, the datastore contents are replicated between Exceed TurboX Servers.

General data, such as profiles and session information, is stored in `etxdata`. Binary files, such as runtimes and screen shots, are stored directly in the file system.

> **Note**
>
> The `data` folder must have permissions that match the user running `etxsvr`. It is recommended that you extract Exceed TurboX Server to disk using the same user that will run `etxsvr`.

For information about backing up the datastore, see the Exceed TurboX Server Manager Help.

> **Note**
>
> To manually back up to a directory, you must first stop the `etxsvr` process with `etxsvr stop`. Backups of a standalone Exceed TurboX Server should be created with the `etxsvr datastore backup` command, which can be run on a live server. To back up a High Availability Exceed TurboX Server Cluster, see the *Exceed TurboX High Availability Configuration Guide* for the recommended backup and restore procedures.

Exceed TurboX Server

Exceed TurboX Server is the intermediary component that communicates with user workstations and with Exceed TurboX Connection Nodes.

This Web application processes user requests, and allows users to access the following applications on the Web:

- **Dashboard**: Web-based application used by Exceed TurboX users to create and manage profiles, configure user settings, launch and manage sessions, and view messages sent by administrators. For detailed information about this interface, see the Exceed TurboX Dashboard Help.

- **REST APIs**: Web-based application used by users and server administrators to navigate through and interact with the Server REST APIs provided with the product. For detailed information, see the *Exceed TurboX Dashboard Help* and *Exceed TurboX REST API User Guide*.
- **Server Manager**: Web-based application used by administrators to remotely configure and administer Exceed TurboX. For detailed information about this interface, see the Exceed TurboX Server Manager Help.

You can access these web applications from any computer that has a supported web browser. For information about the web browsers supported by Exceed TurboX Server Manager, see the *Exceed TurboX Release Notes*.

About High Availability

High Availability refers to a system or component that is continuously available for a long period of time. The goal of high availability solutions is to reduce or eliminate downtime.

Exceed TurboX Servers support high availability configuration on Linux and Solaris platforms only. In supported environments, multiple Exceed TurboX Servers can be used to create an Exceed TurboX Server cluster for high availability (HA cluster), load balancing, and scalability. All Exceed TurboX Servers in the cluster can be configured to run as active servers. All Exceed TurboX users can connect to any of the Exceed TurboX Servers in the cluster to access Exceed TurboX Dashboard, Server Manager, and REST APIs. For detailed information about how to install and configure the Exceed TurboX cluster, see *Exceed TurboX High Availability Configuration Guide*, version 12 (or later).

> **Note**
>
> Support for Solaris was added to Exceed TurboX 12, Service Pack 3.

For more information about configuring an HA cluster, see *Exceed TurboX High Availability Configuration Guide*.

Best Practices for Optimizing Performance

Use these best practices to optimize the performance of Exceed TurboX (ETX) sessions.

ETX performance involves three main components and the network between them: the application, the ETX node that launches a per-session etxproxy process, and the ETX client on the user's computer.

ETX session performance is affected by several factors:

- Network latency and bandwidth
- Architecture
- GPU availability
- CPU speed

Network latency and bandwidth

The network latency between the node host and the ETX user's computer has the most significant effect on session performance.

Latency impacts application responsiveness and directly affects Bandwidth Delay Product (the amount of data that can be in transit in the network). Although you will likely not have very much control over network latency, make sure that:

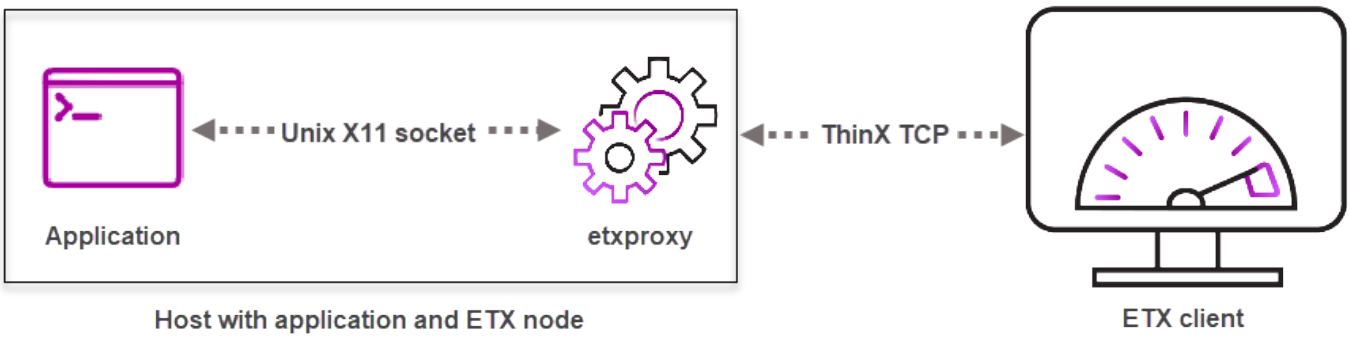
- Your routing is as efficient as possible.
- No network consolidators are shaping ETX traffic. ETX performs its own bandwidth management and requires unmanipulated connections to optimize data flow. ETX traffic flows via TCP on the node port chosen at install time (the default is 5510).

Architecture

You can significantly improve performance by installing the ETX node that starts the etxproxy on the host that runs the user applications instead of installing it on a separate host.

Install the ETX node on the same host that runs applications

Installing the ETX node that starts the etxproxy on the host that runs user applications is one of the most critical factors for improving performance. Applications run faster and with lower resource usage when the etxproxy and the application are on the same host.



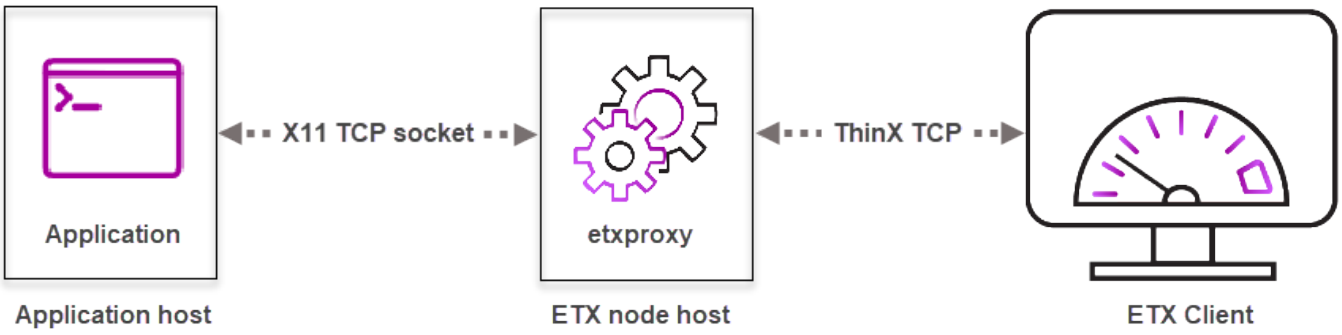
Installing the ETX node on the same host as the application improves performance

When applications are on the same host as the etxproxy, a Unix socket provides efficient and fast communication between the applications and the etxproxy.

> **Note**
>
> ETX always uses the custom ThinX protocol between the etxproxy and the ETX client. The X11 protocol used by applications to communicate with ETX was originally designed for lower-latency networks and applications with lower graphical demands. ETX uses the ThinX protocol to accelerate the high graphical loads and many round-trip queries of modern applications.

Avoid installing the ETX node on a different host than the applications

When applications are not on the same host as the node that starts the etxproxy, communication between the applications and the etxproxy is slower and less efficient. This architecture forces SSH tunneling of the raw X11 protocol between the hosts, removing many of the benefits ETX can provide. It adds latency, can saturate the data center network, adds resource usage, and adds complexity in the SSH tunnel.



Installing the ETX node on a different host than the application hinders performance

The exact performance difference between these architectures is highly dependent on how the application draws.

GPU availability

Some graphical output compresses well with video codecs. If a supported Graphics Processing Unit (GPU) is available, ETX uses it to decrease CPU encoding and network load. This can significantly boost performance for this type of graphical output.

To determine if a graphical load is suitable for video compression:

1. Use the **Performance** Panel to set a very low **Target bandwidth**, e.g. 0.3Mbps.
2. Observe any changes in the Codec usage graph.

If H.264-sw (software) is used, adding a GPU will allow H.264 hardware to be used instead. See GPU support for more details.

CPU speed

ETX uses the CPU to render application drawing commands and compress graphics to send to the ETX client on the user's computer.

The CPU on the ETX node host is unlikely to affect performance. Unless the Performance panel indicates the etxproxy has a high CPU load, moving from an average CPU to a faster CPU will not likely make a difference in user-perceived performance.

ETX takes advantage of AVX2 support.

Some loads, like software OpenGL rendering, benefit greatly from more CPUs. Other loads don't use additional CPUs.

Profile settings

Make sure any Xstart profiles are using the Local X Window Host Application startup method and not SSH, keeping communication over a direct unix socket.

Most applications perform better with MIT-SHM enabled. (This setting is in the profile's **Advanced mode**, on the **Protocol** tab.) This increases application-to-etxproxy performance, and also allows better caching.

Launch applications directly in Multiple Window Mode rather than using Single Window Mode with a full remote desktop environment. Desktops like GNOME 3 significantly increase the drawing and bandwidth required because they do remote compositing. If a remote desktop environment is required, Xfce with compositing disabled provides better performance. To ensure compositing is disabled for both older and newer Xfce versions:

- Uncheck Xfce's **Enable display compositing** setting under **Applications** → **Settings** → **Window Manager Tweaks** on the **Compositor** tab. This setting can also be disabled from the command line via `xfconf-query -c xfwm4 -p /general/use_compositing -t bool -s false`.
- Uncheck the **Composite** extension in the ETX profile's **Advanced mode**, on the **Protocol** tab.

Exceed TurboX GPU support

Exceed TurboX uses hardware Graphics Processing Unit (GPU) acceleration.

When Does ETX Use a GPU?

ETX makes use of hardware GPU acceleration in a few scenarios, based on the node platform and where the application runs.

Session Type	Application Drawing Uses GPU	Session Encoding Uses GPU	GPU Balancing Type
Windows Direct	Yes (OpenGL, D3D, Vulkan.)	Yes[1] [2]	Windows Round Robin
Linux node X session	Yes (Using ssrrun or HISSR. OpenGL only, no Vulkan)	Yes[1] [2]	X Load
Windows node X session	No	Yes[1] [2]	Windows Round Robin
AIX/Solaris node X session	No	No	X Load

[1] ETX 11-12.0.4 supports NVIDIA only. From ETX 12.5 we also support AMD (RX 6000 series tested) and Intel (600 series or higher) on Windows.

[2] Session encoding with hardware H264 will be automatically selected only if the type of drawing and current network conditions will benefit from it. AMD and Intel do not support 4:4:4 (no colour subsampling) because of hardware limitations, and will not be used on content that appears to need 4:4:4. Override this with proxy.Subsample=2

How Does ETX Choose a GPU?

When a system has multiple GPUs available, ETX tries to balance the load between them. There are two mechanisms used to choose the GPU: Windows systems use Windows Round Robin, while Linux or Unix systems use X Load.

Windows Round Robin

This mechanism is built into Windows and controlled by Microsoft. It is session based, not load based. This means Windows session 1 gets GPU 1, Windows session 2 gets GPU 2, and so on. For example, with three GPUs:

```
Session 1 -> GPU1
Session 2 -> GPU2
Session 3 -> GPU3
Session 4 -> GPU1
Session 5 -> GPU2
Session 6 -> GPU3
Session 7 -> GPU1
...
```

Choosing a GPU always requires making a prediction about future needs. Session-based balancing assumes that every user's average GPU load will be roughly equal in the future, regardless of current momentary load, and gives the optimal result for that assumption.

This load balancing only works on Windows Server 2019 or later, and has the following requirements:

1. Group policy must be set to enable balancing. ETX Node install configures this by default, but an admin can choose not to. Verify the current setting with `gpedit.msc` and look under:

   `Local Computer Policy/Computer Configuration/Administrative Templates/Windows Components/Remote Desktop Services/Remote Desktop Session Host/Remote Session Environment/Use the hardware default graphics adapter for all Remote Desktop Services sessions`

2. KB5012636 or an equivalent patch level must be installed. It is included in the latest cumulative update for Server 2019 or Server 2022.

X Load

This mechanism is part of ETX and is used with Server-side OpenGL rendering via the `ssrrun` script, which calls `getssrdipslay.sh` in the node installation to determine the GPU with lowest load at that moment. This script weights rendering, encode and memory utilization. Because this decision is made at the moment the application launches, it assumes future GPU loads will continue to look like they do now. Users may customize the particular algorithm in that script.

- When the `+v` argument is used with `ssrrun`, the selected GPU name, display and load are printed.
- Currently works only with NVIDIA GPUs.
- Assumes GPUs are assigned separate screens on `display :0`. This is the default configuration of `nvidia-xconfig --enable-all-gpus`) If `-d :n` is used, it will use screens on display `n`.

Notes

- ETX-6303 On Linux the first GPU is always chosen for video encode.
- ETX-21829 If the Windows system is unable to share video surfaces then all video encoding is done with the first GPU.

Which GPUs Does ETX Support?

For OpenGL server-side rendering any compliant OpenGL GPU and driver that works with X.org is sufficient. In practice NVIDIA workstation or datacenter GPUs are most tested.

For video encoding, ETX supports encoding with NVIDIA, AMD and Intel GPUs.

Feature	AMD	NVIDIA	Intel
H.264 4:2:0	✓	✓	✓

Feature	AMD	NVIDIA	Intel
H.264 4:4:4	✗	✓	✗

Using GPUs for OpenGL Server-Side Rendering on Linux

An application's OpenGL drawing performance and capability depends on whether a CPU or GPU does the rendering and whether the connection to this renderer is direct or indirect (translated to network protocol between application and renderer). Default rendering is typically indirect via CPU which is more flexible and broadly available, but ideal performance and capability come from direct rendering by a GPU, called Direct Server-Side Rendering or DSSR. DSSR allows multiple users to share a single GPU for 3D rendering. If multiple GPUs are installed, applications are automatically load balanced between them (see X Load).

Requirements

Using DSSR requires the following on the remote application host:

- A supported GPU and driver must be installed.
- `ssrconfig` script must be run, either by the node installer or manually after video driver is installed, so that an X server is enabled and accessible by the end user.
- The application must be launched with the `ssrrun` wrapper script found in the node installation (e.g. `/opt/etxcn/bin/ssrrun myapp`).

Troubleshooting

All these commands must be run on the application host where the connection node and GPU are installed.

Is a supported GPU with an OpenGL driver installed?

- Any GPU with compliant OpenGL drivers ought to work.
- In practice, NVIDIA GPUs with the NVIDIA binary driver are most tested.

Check if a GPU is connected to the host:

```
$ lspci | grep VGA
0b:00.0 VGA compatible controller: NVIDIA Corporation GK106GL [Quadro K4000] (rev
a1)
```

Check if the driver is installed. The grep results above show an NVIDIA GPU, so `nvidia-smi` is used.

```
$ nvidia-smi
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.223.02   Driver Version: 470.223.02   CUDA Version: 11.4     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Quadro K4000        Off  | 00000000:0B:00.0 Off |                  N/A |
| 30%   33C    P8    10W /  87W |     13MiB /  3018MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A      2359      G   /usr/libexec/Xorg                   7MiB |
|    0   N/A  N/A      3768      G   /usr/bin/gnome-shell                1MiB |
+-----------------------------------------------------------------------------
+
```

Some NVIDIA drivers may require `/etc/pam.d/exceed-connection-node` PAM configuration. New node installers create this but for existing installations where the file doesn't already exist, consult the documentation for details on creating this file or simply run `cp /etc/pam.d/sshd /etc/pam.d/exceed-connection-node`.

Is the X server set up and accessible by the end user?

If the video driver is installed after the node, manually run the `ssrconfig` script

```
/opt/etxcn/bin/ssrconfig config
```

Reboot after you run the script to ensure the new settings take effect.

The node installer asks to run the `ssrconfig` script when the node is installed but you'll need to run it again after installing the video driver.

This script configures several things, including:

- Switching the default graphical server from Wayland to X
- Switching the system to graphical login, so the X server starts with the system
- Ensuring the login manager gives users access to the X server
- Setting device permissions so all users can access the GPU
- Installing custom VirtualGL libraries to the system lib directory

Verify X server is running

```
$ ps -ef | grep Xorg
root        1952    1946  0 09:53 tty1     00:00:00 /usr/libexec/Xorg vt1 -
displayfd 3 -auth /run/user/42/gdm/Xauthority -nolisten tcp -background none -
noreset -keeptty -novtswitch -verbose 3
```

If Xorg isn't running, check the following:

- Is the display manager running? On many systems, you can use the following command to check: `systemctl status display-manager`
- Did Xorg start successfully? Check `/var/log/Xorg.0.log` for EE errors.
- For NVIDIA errors like *(EE) No devices detected*, try running `ssrgennvidiaxorgconf`. For example:

```
/opt/etxcn/bin/ssrgennvidiaxorgconf
```

The following commands assume Xorg is running on DISPLAY `:0`. Depending on the system, the display number might be different.

Verify access

`xdpyinfo -display :0`

- A successful call will produce many lines of output describing the display.
- If a *No protocol specified* error is returned, it means the user does not have permissions that should have been set up by `ssrconfig`.
  For temporary testing, these permissions can be added manually:

```
$ ps -ef | grep Xorg
root        1952    1946  0 09:53 tty1     00:00:01 /usr/libexec/Xorg vt1
-displayfd 3 -auth /run/user/42/gdm/Xauthority -nolisten tcp
-background none -noreset -keeptty -novtswitch -verbose 3
$ sudo su
# xauth merge /run/user/42/gdm/Xauthority
# DISPLAY=:0 xhost +LOCAL:
non-network local connections being added to access control list
# exit
$ xdpyinfo -display :0
```

Verify GPU is used by the X server

```
$ glxinfo -display :0 | grep OpenGL.renderer
OpenGL renderer string: Quadro K4000/PCIe/SSE2
```

- If the returned string is *llvmpipe*, the X server is not using the GPU for OpenGL rendering. Examine `/var/log/Xorg.0.log` for clues and consider re-installing the GPU driver.
- If the expected GPU name is included in the renderer name, DSSR should work.

Does DSSR work?

Launch a profile and open a terminal, then run the following command:

```
$ /opt/etxcn/bin/ssrrun -logo /opt/etxcn/3rdparty/virtualgl/bin/glxspheres64
GLXSpheres version 1.0
Polygons in scene: 62464 (61 spheres * 1024 polys/spheres)
GLX FB config ID of window: 0xdd (8/8/8/0)
Visual ID of window: 0x21
Context is Direct
OpenGL Renderer: Quadro K4000/PCIe/SSE2
151.473167 frames/sec - 169.044055 Mpixels/sec
```

- `ssrrun` redirects OpenGL rendering commands to the local X server with GPU attached, delivering the final output image to the user.
- Note the context is direct and renderer is the expected GPU.
- Expect a window with colorful rotating spheres to appear. Examine the lower right corner of the window and expect overlaid letters `VGL`, which indicate DSSR is operating successfully.
- Running `nvidia-smi` in another terminal will show that `glxspheres64` is using the GPU.

3. Installing Exceed TurboX Server components

Installing Exceed TurboX Server components

This section describes how to install the Exceed TurboX Server and Connection Nodes.

This section includes the following topics:

- Installing Exceed TurboX Server
- Installing Exceed TurboX Connection Nodes
- Silently installing Exceed TurboX Nodes

Installing Exceed TurboX Server

Installing Exceed TurboX Server

This section describes how to install Exceed TurboX Server as a Stand-alone server on Linux or UNIX platforms, and Windows platforms. It also provides information about how to run, stop, check the status of, and configure the settings of, your Exceed TurboX Server. For detailed information, see the following topics:

Linux and UNIX platforms:

- Installing Exceed TurboX Server on a Linux or UNIX platform
- Starting and stopping Exceed TurboX Server on a Linux platform
- Configuring Exceed TurboX Server settings

Windows platforms:

- Installing Exceed TurboX Server on a Windows platform
- Starting and stopping Exceed TurboX on a Windows platform
- Configuring Exceed TurboX Server settings

> **Important**
>
> The Exceed TurboX Server contains an embedded datastore and must be installed on a locally attached disk for performance and stability reasons. If the Exceed TurboX Server is installed on a network attached storage (NAS) volume, such as NFS, a brief disconnection of the NAS can result in data corruption and downtime.

For information about how to install and configure Exceed TurboX Servers in High Availability environments (Linux and Solairs platforms only), see About High Availability.

Installing Exceed TurboX Server on Linux

This section describes how to install and start Exceed TurboX Server as a Stand-alone server on Linux.

The following procedure describes the Server installation performed by the `root` user. The installation can be performed by a non-root user, but certain limitations will apply. The Exceed TurboX Server process will run as the user who performed the installation. Once the Server is installed and run one time, the installation is permanently tied to the user who installed the Server. The user that installs the Exceed TurboX Server owns the installation folder and `etxdata` content. The owner of the installation cannot be changed. (For example, running the `chown` command will not change the owner of the installation, even if the Server failed to start.

The following limitations apply if a non-root user performs the installation:

- A non-root user must use an HTTP/HTTPS port number above 1023.
- System resources for non-root users may be limited compared to resources for the `root` user. Unless these limitations are removed from the system settings, performance may be limited when running a Server under a regular user.

The Exceed TurboX Server contains an embedded datastore and must be installed on a locally attached disk for performance and stability reasons. If the Exceed TurboX Server is installed on a network attached storage (NAS) volume, such as NFS, a brief disconnection of the NAS can result in data corruption and downtime.

The *To install Exceed TurboX Server on Linux or UNIX platforms* procedure is almost the same as the *To migrate Exceed TurboX Server (on Linux or UNIX platforms)* procedure. Consider sharing this content as a file entity in the next release, if the time allows for this change.

To install Exceed TurboX Server on a Linux or UNIX platform:

1. Connect to the Server host (for example, using SSH) as `root` (recommended) to open a remote terminal.
2. Locate the installation media folder, whether it is in a local disk or on the network.
3. Create a directory called `etxsvr` for the installation files:

   ```
   sudo mkdir /opt/etxsvr
   ```

4. Type your password at the prompt.

   If you have permission to use the `sudo` command, this creates an `etxsvr` directory in the root of the file system.

   If you do not have permission to use `sudo`, contact your system `sudo` administrator.

5. Go to the directory you created:

   ```
   cd /opt/etxsvr
   ```

6. Extract the files from the installation package into the `/opt/etxsvr` directory:

   ```
   sudo tar xzvf path/packagename.tar.gz
   ```

   where:

   - `path` is the path to the `ETXServer` directory on the product installation media.
   - `packagename` is the installation file corresponding with your system (`ETXServer-ETX Version.Build Number-SP Number-Platform-CPU.tar.gz`).

     The following example is the Linux package name for version 12.0.1, build 5790:

     ```
     ETXServer-12.0.1.5790-SP1-linux-x64.tar.gz
     ```

7. Enter your password, if you are prompted.

   The Exceed TurboX Server is now installed and ready to start. Do one of the following:

   - To start the Server using the default pre-configured settings provided with the product, see Starting and stopping Exceed TurboX Server on a Linux platform.
   - To customize the Server settings before starting the Server, see Configuring Exceed TurboX Server settings.

Installing Exceed TurboX Server on a Windows platform

This section describes how to install and start Exceed TurboX Server as a Stand-alone server on a Windows platform, and how to apply a service pack update. Installing a service pack update utilizes the same procedure as installing the Exceed TurboX Server.

The following procedure describes the Server installation performed by the Administrative user. The Exceed TurboX Server process is intended to run as the Administrative user that is selected during the installation process. Once the Server is installed and run at least one time, the installation is permanently tied to this Administrative user. The user who installs the Exceed TurboX Server owns the installation folder and the etxdata content. The owner of the installation cannot be changed.

The Exceed TurboX Server contains an embedded datastore and must be installed on a locally attached disk for performance and stability reasons. If the Exceed TurboX Server is installed on a network attached storage (NAS) volume, such as NFS, a brief disconnection of the NAS can result in data corruption and downtime.

You can install one instance of Windows Exceed TurboX Server per machine.

To install Exceed TurboX Server on a Windows platform:

1. Copy the Windows install package to your Windows server.
2. Run one of the following:
   - `OpenTextETXServer<version>.msi`
   - `setup.exe`
3. You will be prompted to read the **End User License Agreement**. To continue with the installation, select **I accept the terms of the License Agreement**, and then click **Next**.
4. In the Setup Wizard, click **Next** to accept the default path for the installation folder. Or, click **Change** to change it.

   The default folder path is:

   `C:\Program Files\OpenText\ETXServer\`
5. Specify the user account that will be used as the Administrative account for Exceed TurboX Server, then click **Next**.

   The account defined here will be the owner of the Exceed TurboX installation folder and the `etxdata` content.
6. Click **Install**.

   The installation process runs. An **Exceed TurboX Server Updates** window displays the installation progress.
7. Click **Finish** when prompted.

The installation process is complete. You can now start Exceed TurboX Server.

To start the Server using the default, pre-configured settings provided with the product, see Starting and stopping Exceed TurboX on a Windows platform.

To customize the Server settings before starting the Server, see Configuring Exceed TurboX Server settings.

Starting and stopping Exceed TurboX Server on Linux

This section describes tasks you can perform to maintain the Server for proper operation on Linux platforms.

You can perform the following tasks to maintain the Exceed TurboX Server:

- Check the status of the Server.

  Exceed TurboX Dashboard and Server Manager are not accessible when the Server is not running. You may need to check the status of the Server if you are unable to access the Dashboard or Server Manager.

- Start the Server.

  You can run the Exceed TurboX Server as an application or as a service. Starting the Server as a service allows sessions to keep running, even if the session owner signs off from the session where the Exceed TurboX Server is launched. It also allows the Exceed TurboX Server to start when the machine starts (the computer on which the Exceed TurboX Server is installed). Running the Server as an application ties the state of the Dashboard and Server Manager with the user's session. However, running the Server as an application is useful for short-term tests and configuration.

- Stop the Server.

  The Server must be stopped before a patch can be applied.

To check the status of the server:

1. Open a Terminal window by selecting **Applications** > menu > **System Tools** > **Terminal**.
2. Change directory to the Exceed TurboX Server installation folder:

   ```
   cd /opt/etxsvr/etxsvr-version
   ```

3. Do one of the following:

   - If the server is running as an application, enter the following command:

```
bin/etxsvr status
```

- If the server is running as a service, enter one of the following commands:

  ```
  systemctl status otetxsvr
  ```

  or

  ```
  systemctl status otetxsvr_n
  ```

To start the Server as a service:

1. If the Server is running as an application, stop the Server before starting it as a service.
2. Open a Terminal window by selecting **Applications** > menu > **System Tools** > **Terminal**.
3. Change directory to the Exceed TurboX Server installation folder.

   ```
   cd /opt/etxsvr/etxsvr-version
   ```

4. Enter the following command to create the systemd scripts:

   ```
   bin/etxsvr bootstart enable
   ```

   The command prints the name of the systemd service it created. For example, `otetxsvr`.

   If there is more than one instance of extsvr on the computer, the name of the service has the format `otetxsvr_n`, where `n` is a unique identifying number. For example, `otetxsvr_2`.

5. Enter the following command to start the service:

   ```
   systemctl start otetxsvr
   ```

   or

   ```
   systemctl start otetxsvr_n
   ```

6. To open the Exceed TurboX Server Manager, type `https://<servername>:<port>/etx/admin` into a web browser window.

   > **Tip**
   >
   > If you start the Exceed TurboX Server as an application, the URL will be displayed in the command prompt window. To get the URL, you can start the Server as an application, stop it, and restart it as a service.

7. If you are starting the Server for the first time, you will be prompted to review the **End User License Agreement**. If you click **Accept**, you will be prompted to create a password for the etxadmin user.

   Once you create the password, you will be able to sign in to the ETX Server Manager and continue with Server configuration. For more information, see *Signing in to Exceed TurboX using the etxadmin account* and *Signing in to Exceed TurboX Server Manager* in *Exceed TurboX Server Manager Help*.

To start the Server as an application:

1. Open a Terminal window by selecting **Applications** > menu > **System Tools** > **Terminal**.
2. Change directory to the Exceed TurboX Server installation folder.

   ```
   cd /opt/etxsvr/etxsvr-version
   ```

3. Enter the following command:

   ```
   bin/etxsvr start
   ```

   The Server starts and the URL to connect to the Server Manager is displayed.
4. Copy the URL and paste it into a Web browser window to open the Server Manager.
5. If you are starting the Server for the first time, you will be prompted to review the **End User License Agreement** >. If you click **Accept**, you will be prompted to create a password for the etxadmin user.

   Once you create the password, you will be able to sign in to the ETX Server Manager and continue with Server configuration. For more information, see *Signing in to Exceed TurboX using the etxadmin account* and *Signing in to Exceed TurboX Server Manager* in *Exceed TurboX Server Manager Help*.

To stop the Server:

1. Open a Terminal window by selecting **Applications** > menu > **System Tools** > **Terminal**.
2. Change directory to the Exceed TurboX Server installation folder.

   ```
   cd /opt/etxsvr/etxsvr-version
   ```

3. Do one of the following:

   - If the Server is running as an application, enter the following command:

     ```
     bin/etxsvr stop
     ```

   - If the Server is running as a service, enter the following command:

     ```
     systemctl stop otetxsvr
     ```

     or

     ```
     systemctl stop otetxsvr_n
     ```

> **Note**
>
> When the server is stopped, the Dashboard and Server Manager are not accessible. Any existing sessions will continue to run but users will not be able to manage them using the Dashboard.

Starting and stopping Exceed TurboX on a Windows platform

This section describes tasks you can perform to maintain the Server for proper operation on Windows platforms.

You can perform the following tasks to maintain the Exceed TurboX Server:

1. Check the status of the Server.

   Exceed TurboX Dashboard and Server Manager are not accessible when the Server is not running. You may need to check the status of the server if you are unable to access the Dashboard or Server Manager.

2. Start the Server.

   You can run the Exceed TurboX Server as an application or as a service. Starting the Server as a service allows sessions to keep running, even if the session owner signs off from the Windows session where the Exceed TurboX Server is launched. It also allows the Exceed TurboX Server to start when the machine starts (the computer on which the Exceed TurboX Server is installed). Running the Server as an application ties the state of the Dashboard and Server Manager with the user's Windows session. However, running the Server as an application is useful for short-term tests and configuration.

   You can start the Server as an application, even if you configure etxsvr to run as a service. See Configuring Exceed TurboX Server settings for more information about configuring the Server.

3. Stop the Server

   The Server must be stopped before a patch can be applied.

To check the status of the server:

1. Open the **Administrative Command Prompt for ETX Server** shortcut.

   This opens a Windows command prompt in the Exceed TurboX Server home directory.

2. Enter the following command:

   ```
   bin\etxsvr status
   ```

To start the Exceed TurboX Server as a service:

1. If the Server is running as an application, stop the Server before starting it as a Service.
2. Open the **Administrative Command Prompt for ETX Server** shortcut.

   This opens a Windows command prompt in the Exceed TurboX Server home directory.

3. Enter the following command:

   ```
   bin\etxsvr bootstart enable
   ```

You are prompted to enter the password of the user that installed and owns the Server.

4. Enter the password of the user that owns the Server.
5. Enter the following command to start the Server as a service:

   ```
   net start etxservice
   ```

6. To open the Exceed TurboX Server Manager, type `https://<servername>:<port>/etx/admin` into a web browser window.

   > **Tip**
   >
   > If you start the Exceed TurboX Server as an application, the URL will be displayed in the command prompt window. To get the URL, you can start the Server as an application, stop it, and restart it as a service.

7. If you are starting the Server for the first time, you will be prompted to review the **End User License Agreement**. If you click **Accept**, you will be prompted to create a password for the etxadmin user.

   Once you create the password, you will be able to sign in to the Exceed TurboX Server Manager and continue with Server configuration. For more information, see *Signing in to Exceed TurboX using the etxadmin account* and *Signing in to Exceed TurboX Server Manager* in *Exceed TurboX Server Manager Help*.

To start the Exceed TurboX Server as an application:

1. Open the **Administrative Command Prompt for ETX Server** shortcut.
2. Run the following command:

   ```
   bin\etxsvr start
   ```

   The Server starts and the URL to connect to the Server Manager is displayed.
3. Copy the URL and paste it into a Web browser window to open the Server Manager.
4. If you are starting the Server for the first time, you will be prompted to review the **End User License Agreement**. If you click **Accept**, you will be prompted to create a password for the etxadmin user.

   Once you create the password, you will be able to sign in to the Exceed TurboX Server Manager and continue with Server configuration. For more information, see *Signing in to Exceed TurboX using the etxadmin account* and *Signing in to Exceed TurboX Server Manager* in *Exceed TurboX Server Manager Help*.

To stop the server:

1. Open the **Administrative Command Prompt for ETX Server** shortcut.
2. Enter the following command:

   ```
   bin\etxsvr stop
   ```

> **Note**
>
> When the server is stopped, the Dashboard and Server Manager are not accessible. If the Server was running as a service, any existing sessions will continue to run but users will not be able to manage them using the Dashboard.

Configuring Exceed TurboX Server settings

You can configure Exceed TurboX Server settings by running the `bin/etxsvr config key=value` command.

This section provides a list of settings you can use to configure the Exceed TurboX Server on Linux, UNIX, and Windows platforms.

Key	Value
dashboardHttpEnable	Set to `1` to enable communication using unencrypted (non-TLS) HTTP connections for Exceed TurboX Dashboard and REST APIs. Default value is `1`.
dashboardHttpsEnable	Set to `1` to enable communication using TLS-encrypted HTTPS connections for Exceed TurboX Dashboard and REST APIs. Default value is `1`.
adminHttpEnable	Set to `1` to enable communication using unencrypted (non-TLS) HTTP connections for Exceed TurboX Server Manager. Default value is `1`.
adminHttpsEnable	Set to `1` to enable communication using TLS-encrypted HTTPS connections for Exceed TurboX Server Manager. Default value is `1`.
dashboardHttpPort	When **dashboardHttpEnable** is set to `1`, specify the port number to be used by Exceed TurboX Dashboard and REST APIs for unencrypted (non-TLS) HTTP connections. Default value is `8080`.
dashboardHttpsPort	When **dashboardHttpsEnable** is set to `1`, specify the port number to be used by Exceed TurboX Dashboard and REST APIs for TLS-encrypted HTTPS connections. Default value is `8443`.
adminHttpPort	When **adminHttpEnable** is set to `1`, specify the port number to be used by Exceed TurboX Server Manager for unencrypted (non-TLS) HTTP connections. Default value is `8080`.

Key	Value
adminHttpsPort	When **adminHttpsEnable** is set to `1`, specify the port number to be used by Exceed TurboX Server Manager for TLS-encrypted HTTPSconnections. Default value is `8443`.
httpToHttpsRedirection	Set to `1` to redirect all HTTP requests (received on the specified HTTP port) to HTTPS (to the specified HTTPS port). This ensures that all connections to the Exceed TurboX Server web interfaces are encrypted with SSL/TLS, and that users who try to access the Server using HTTP arrive at the secure HTTPS sign-in page, rather than seeing a "404" error. Default value is `0`.
	Note: REST API HTTP connections are not redirected to HTTPS, because some clients may not handle them correctly, if redirected.
httpsCipherSuites	Overrides cipher suites used for HTTPS connections. The default value is `0` (do not override). If `0` is used, Exceed TurboX Server uses a default list defined by Exceed TurboX Server and supported by Java.
	The valid format is a comma-separated list of cipher suites, listed in the order in which they will be used by Exceed TurboX Server for all HTTPS connections.
logLevel	Specify the level of detail to be included in Exceed TurboX Server logs. Possible values are: `TRACE`, `DEBUG`, `INFO` (default value), `WARN`, and `ERROR`.
	Note: Administrators can view and download the Server logs on the **Server Manager** > **Server Settings** > **Server Logs** page. See "Reviewing Server logs" in the *Exceed TurboX Server Manager Help* for more information.
logFilter	Specify a log filter that the Exceed TurboX Server can use to collect additional information for troubleshooting server issues. You should set the log filter only at the direction of customer support.
accessLogEnable	Set to `1` to instruct Exceed TurboX Server to collect information about web access to the Server logs. Default value is `0`.

Key	Value
	Note: You can view Server logs on the Server Manager > **Server Settings** > **Server Logs** page. See "Reviewing Server logs"in the *Exceed TurboX Server Manager Help* for more information.
accessLogRetainDays	Specify the number of days to keep access log files on the Server. This setting determines the number of access log files made available for review on the **Server Settings** > **Server Logs** page. Default value is `7`.
eulaAccepted	Set to `1` to accept the terms of the **End User License Agreement** and bypass this page when signing in to the Exceed TurboX Server web interface. Default value is `0`.
maintenanceMode	Set to `1` to switch Exceed TurboX Server to maintenance mode. Default value is `0`.
maintenanceModeMessage	Specify the message to be displayed to Exceed TurboX users while the Server is in maintenance mode.
maxJavaHeapSize	Specify the maximum size of the Java heap as a percentage of total system memory. Default value is `50%`.
etxdbShmSize	Specify the maximum value of the memory allocated for the etxdb process, in MB. Default value is `128 MB`.

Installing Exceed TurboX Connection Nodes

Installing Exceed TurboX Connection Nodes

This section describes how to install Exceed TurboX Connection Nodes on UNIX-based and Windows platform.

A Connection Node is the component that provides the X server engine, Proxy Manager, and the Authenticator. For more information about Nodes, see About Exceed TurboX server-side components.

> **Caution**
>
> This is a mandatory step in the server-side installation process for all authentication methods except LDAP, OTDS and WAM. A Connection Node that is given the Authentication role is required to allow for these types of authentication methods.

This section contains the following topics:

Installing Exceed TurboX Connection Nodes on Linux

This section describes how to install Exceed TurboX Connection Nodes on Linux.
This is a mandatory step in the server-side installation process for all authentication methods except LDAP, OTDS, and WAM. In a single-node environment, the node serves as the Proxy Manager and as the Authenticator.
In an environment with multiple Connection Nodes, each node must be installed individually.

> **Important**
>
> ETX Nodes must be installed as root.

To install an Exceed TurboX Connection Node:

1. Connect to the Node host (for example, using SSH) and become `root` to open a remote terminal.
2. Locate the installation media folder, whether it is in a local disk or on the network.
3. Create a directory called `etxcn` for the installation files:

   ```
   sudo mkdir /opt/etxcn
   ```

4. Type your password at the prompt.

   If you have permission to use the `sudo` command, this creates an `etxcn` directory in the root of the file system.

   If you do not have permission to use `sudo`, contact your system `sudo` administrator.

5. Go to the directory you created:

   ```
   cd /opt/etxcn
   ```

6. Extract the files from the installation package into the `/opt/etxcn` directory:

   `sudo tar xzvf path/packagename.tar.gz`

   where:

   - `path` is the path to the `ETXConnectionNode` directory on the product installation media.
   - `packagename` is the installation file corresponding with your system (`ETXCN-ETX Version.Build Number-SP Number-Platform-CPU.tar.gz`).

     The following example is the Linux package name for version 12.5.3, build 9328: `ETXCN-12.5.3.9328-linux-x64.tar.gz`

7. Run the following installation script:

   `sudo etxcn/bin/install`

   Enter your password if you are prompted. The installation script starts.

8. When prompted, press **Enter** to indicate you are not migrating from an existing installation.
9. When prompted, type `y` to create the `etxproxy` and `etxstart` fallback accounts. The default value is `n`.

   > **Note**
   >
   > The `etxproxy` and `etxstart` accounts are local fallback accounts. By default, these accounts are used to run the proxy and Xstart processes, respectively, if the user signed in does not have a local account. For information on the effects of enabling fallback accounts, see "Understanding Exceed TurboX fallback accounts" in the *Exceed TurboX Server Manager Help*.

   When you select whether to create a fallback account, the installer responds as shown below:

   The `etxproxy` account

   `Y` The installer configures and creates the `etxproxy` account and checks directory access.

   `N` The `etxproxy` account is not created.

   The `etxstart` account

   `Y` The installer configures and creates the `etxstart` account and checks directory access.
   If the home directory is not configured properly, you may receive a warning and need to create the `/home/etxstart` directory.

   `N` The `etxstart` account is not created.

10. When prompted, specify a Connection Node port number, or press **Enter** to accept the default port.

    This is the port on which the Exceed TurboX Connection Node listens, and through which the proxy transfers session data.

    Notes:

- You must ensure that the port is not in use and is open in the firewall on the system.
- You can manually import an existing SSL certificate by running `/bin/sslcertupdate` when the installation is complete.

11. When prompted, specify the Web Adaptor port number, or press ENTER to accept the default port. This port uses the same port as `etxpm` by default.

    The Web Adaptor port is used for Web Client connections.

12. When prompted to start the connection node at boot time, type:

    - `y` if you want to automatically start the Exceed TurboX Connection Node service whenever your system is rebooted.
    - `n` if you want to manually start the service.

13. Choose whether to configure graphics device permissions for Server-Side Rendering (SSR). Type `y` for Yes or press Enter for No.
14. Choose whether to start the Exceed TurboX Connection Node now. Type `y` for Yes or press Enter for No. If you do not start the Connection Node, the installer instructs you to start and then register the node.
15. To register your node, do one of the following, as appropriate to the circumstances for your system:

    - To register your node now using the installer, type `y`, then continue with the next step.
    - To register your node using either Server Manager or the `nodecmds registernode` command (using the Host name and Port shown on the screen), type `n`. Proceed to register the node using your preferred method. Then, continue with Step 21. For information about registering a node using the Server Manager, see "Registering connection nodes" in the *Exceed TurboX Server Manager Help*.

16. When prompted, enter the Server URL with which you want to register the node.
17. Type `y` to use the displayed hostname or type `n` to provide a different hostname. Then, specify the hostname to use.
18. Provide answers to the following questions:

    Use this node for Native/PAM authentication of Exceed TurboX users?

    Type `y` to designate the node as an Authenticator node, which handles user authentication requests.

    Type `n` to not use the node for Native/PAM authentication of Exceed TurboX users. For more information, see "Configuring node settings" in *Exceed TurboX Server Manager Help*.

    Use this node as a Proxy Manager for remote sessions?

    Type `y` to designate the node as a Proxy Manager node, which will manage sessions.

    Type `n` to not use the node as a Proxy Manager.

    Turn on application scanning?

Type `y` to turn on application scanning, which designates the node as an Application node.

Type `n` to not enable application scanning.

Would you like to provide an alternative node address for server or client connections?

Type `y` to provide an alternative node address for server connections and for client connections.

Press **Enter** for No.

For more information on alternative node addresses, see "Configuring node settings" in *Exceed TurboX Server Manager Help*.

The specified registration options are displayed.

19. Confirm that you want to register the node with the displayed settings by typing `y`, or press **Enter** for No.
20. Specify whether you want to connect to the server using API key authentication by typing `y` for Yes or pressing **Enter** for No.

    The node is now registered with the selected Server.
21. When prompted to enter the username to use to connect to the Exceed TurboX Server, type `etxadmin`. This is the built-in administrator account. This account allows you to perform the initial Server configuration.
22. Type the password for the `etxadmin` account.

    You are now connected to the Server and you can review the node information displayed. The node installation is complete.

Installing Exceed TurboX Connection Nodes on Windows

Installing Exceed TurboX Connection Nodes on Windows

This section describes how to install an Exceed TurboX Connection Node on Windows. Connection Nodes on Windows are capable of performing the Proxy Manager role for a site.

Notes

If you only install Connection Nodes on Windows, you are not able to use PAM Authentication method. PAM Authentication methods require UNIX Connection Nodes.

On Windows platforms, only one version of Exceed TurboX Connection Nodes can be installed at one time on the same machine. Installing Connection Nodes that are version 12.5 or newer automatically updates Connection Nodes that are version 12.0.

To install an Exceed TurboX Connection Node on Windows:

1. If this is a Windows Server system and you want to publish applications on this server to multiple users using Exceed TurboX, install the Remote Desktop Services role and connect the server to a Remote Desktop license host. To do so, see Configuring Remote Desktop Services for Windows Server Connection Nodes.
2. Run the `Msetup.exe` file at the root of the installation media folder.
3. Select **Browse the ETX connection node packages**. An Explorer window opens.
4. Go to the `Windows/x64` or (`Windows/arm64` on `ARM64` platform) folder and launch `Setup.exe` using **Run as Administrator**.

   > **Note**
   >
   > This is only for the Windows `ARM64` platform. You can install either `x64` or `arm64` of the native `ARM64` version of the ETX Connection Node. The installed package determines the node runtime packages, session types, and updates used with this node. For example, if you install the `ARM64` package, you will use the `ARM64` native runtimes for all sessions on this node. To switch packages, you can uninstall the first package, then install the `x64` package, and re-register with the Exceed TurboX site.

5. In the Setup Wizard, on the **Welcome** page, click **Next**.
6. Select a destination folder or keep the default folder. Click **Next**.
7. Enter the Connection Node port number to use, or keep the default port number. Click **Next**.
8. Enter the Web Adaptor port to use, or keep the default port number. Then, click **Next**.
9. If you are installing the Connection Node on a Windows Desktop Operating System (for example, Windows 10), go to Step 13.
10. If you are installing the Connection Node on a Windows Server Operating System, Setup Wizard executes several types of checks to determine the contents of the windows.

    The first check determines whether this server is configured as a Remote Desktop Session Host. The result and a list of Exceed TurboX features that are enabled in this step are displayed on the **Remote Access Configuration - Configure Windows Server to launch remote applications** page.

    Review the information and click **Next**.
11. (*Windows Server acting as a Remote Desktop Host*) The second check determines the group policy option enabled for this server. The result and a list of Exceed TurboX features that are enabled in this step are displayed on the **Remote Access Configuration - Change group policy for remote applications** page.

    - If this check is successful, review the information and click **Next**.

- If the check is unsuccessful, review the information and select **Allow remote start of unlisted programs**. Click **Next**.

  This Group Policy option enables this server to allow unlisted applications to run using RDP, and Exceed TurboX to list or publish applications from this node.

> **Note**
>
> This list is empty if the server does not allow unlisted applications to run according to its RemoteApp Manager settings. See the Windows Server article Configure RD Session Host Server Settings for more information.

12. The third check determines the group policy option for running multiple sessions. The result and a list of Exceed TurboXfeatures that are enabled are displayed on the **Remote Access Configuration - Change group policy for running multiple sessions** page.

    - If the check is successful, review the information and click **Next**.
    - If the check is unsuccessful, review the information and select **Enable multiple sessions per user**. Click **Next**.

      This Group Policy option allows a user to launch multiple sessions on this host.

13. The fourth check determines whether Remote Desktop connections are allowed. The result and a list of Exceed TurboX features that are enabled are displayed on the **Remote Access Configuration - Configure Windows to allow remote connections** page.

    - If the check is successful, review the information and click **Next**.
    - If the check is unsuccessful, select one of the following options, and then click **Next**:
      - **Allow remote desktop connections to this computer**. This option allows any clients to connect remotely to this host.
      - **Only allow connections from clients using Network Level Authentication**. This option allows only NLA clients to connect remotely to this host.

14. The last check determines if the Connection Node can use hardware graphic adapters for rendering remote sessions. If not, slower software rendering will be used. The results are displayed on the **Remote Access Configuration - Configure Windows to use hardware default graphics adapter** page.

    Review the information, and then click **Next**.

    The **Ready to Install the Product** page is displayed.

15. Click **Install**.

The connection node is being installed. This may take several minutes. When the operation is complete, the **Setup Wizard Completed** page is displayed.

16. Do one of the following:

    To register your node using the node registration tool: Select **Run the node registration tool** and click **Finish**.

    The **ETX Connection Node Registration** dialog box is displayed. Use this dialog to register your connection node, as follows:

    a. Specify the Exceed TurboX Server URL and the `etxadmin` user name and password.

    b. Configure any node settings, as necessary. Node settings are described in the *Configuring node settings* topic in the *Exceed TurboX Server Manager Help*.

    c. Click **Register** to finish the node registration process.

    To register your node using the Exceed TurboX Server Manager, select **Register the node manually**.

    You can later sign in to Exceed TurboX Server Manager and register your node using the host and port provided on this page. See *Registering connection nodes* in the *Exceed TurboX Server Manager Help* for more information.

17. Click **Next** to finish the installation process.

Configuring Remote Desktop Services for Windows Server Connection Nodes

This section describes how to configure the Remote Desktop Services (RDS) role on Windows Server nodes. Configuring RDS enables you to do the following:

- Scan installed applications and publish them using Exceed TurboX.
- Host remote desktops for multiple users.

> **Note**
>
> This section applies to Windows Server operating systems. Windows desktop systems (such as Windows 10) cannot be used to publish applications or host multiple simultaneous desktop connections.

To configure Remote Desktop Services on your Windows connection node:

1. Install the Remote Desktop Licensing Host service on a Windows server on your network. This includes purchasing and allocating the appropriate number of Client Access Licenses (CALs) required for your intended use. Use the instructions provided by Microsoft at the following URL:

   https://technet.microsoft.com/en-us/library/cc731765%28v=ws.11%29.aspx

2. Install the Remote Desktop Services role on the server that will host your published applications and RDP desktops. Use the instructions provided by Microsoft at the following URL:

   https://technet.microsoft.com/en-us/library/cc742813(v=ws.11).aspx

3. Continue with the installation of the Exceed TurboX connection node. See Installing Exceed TurboX Connection Nodes on Windows.

Enabling RDP Connections

Enabling RDP connections applies to the following:

- Windows 7, Windows 8, and Windows 10
- Windows Server 2008 R2 / 2012 R2 (administration only) / 2016

Remote Desktop connections must be enabled on your Windows hosts in order to access them using profiles configured to use Windows Desktop - RDP. Use the instructions provided by Microsoft for enabling remote desktop protocol at the following URL:

https://support.microsoft.com/en-ca/help/17463/windows-7-connect-to-another-computer-remote-desktop-connection

> **Note**
>
> Remote desktop connections are enabled automatically when adding the Remote Desktop Services role to a Windows Server operating system. See Configuring Remote Desktop Services for Windows Server Connection Nodes.

Understanding how automatic node updates work

Exceed TurboX Connection Node files are packaged inside some Exceed TurboX Server patches. Applying an update to Exceed TurboX Server automatically updates all Connection Nodes to the same version as Exceed TurboX Server. The automated patching procedure eliminates the need to manually patch each node and ensures that the Connection Nodes always use the same version as Exceed TurboX Server.

Silently installing Exceed TurboX Nodes

Silently installing Exceed TurboX Nodes

This section provides information about how to silently install Exceed TurboX Connection Nodes.

For more information, see the following topics:

- Exceed TurboX Node silent installation on Linux and UNIX systems
- Exceed TurboX Node silent installation on Windows systems

Exceed TurboX Node silent installation on Linux and UNIX systems

Exceed TurboX Node silent installation on Linux and UNIX systems

This section provides information about the Exceed TurboX Node silent installation on Linux and UNIX systems.

> **Note**
>
> The node silent installation requires `sudo` privileges.

To install a node silently, run the following command:

```
sudo bin/install -s fullpath/responsefile
```

The `-s` (silent) parameter requires the path to a response file. The `responsefile` contains responses to installation questions and has a specific format. A sample response file is available in the node tar package (`/conf/response.etxcn.template`). To specify install options for the node, edit the `response.etxcn.template` file in a text editor, then pass the path to the edited file when specifying the `-s` option.

You can configure the following properties in the node response file on Linux and UNIX systems.

Property and Values	Description
Basic options	
`install.etxcn.ListenPort=portnumber`	The port number for the Connection Node to use (default port: `5510`).

Property and Values	Description
install.etxcn.StartNow=1 or 0	When set to 1, starts the Connection Node at the end of the installation (required for automatic node registration).
`install.etxcn.AllowMigrate=1 or 0` and `install.etxcn.MigrateFromThisFolder=full/path/to/old CN folder`	These two properties can be used to migrate from a previous version.
	When `install.etxcn.AllowMigrate` is set to 1, allows you to migrate the node from one version to the next. In this case, you must also set the `install.etxcn.MigrateFromThisFolder=full/path/to/old_folder` property.
	Note: Migrating from an existing node stops the node and terminates all running sessions on the node. If the node from which you are migrating is version 11.5 or later, migrating the node is no longer required. Node updates are pushed out by Exceed TurboX Server after you apply an update to the server.
`install.etxcn.SaveConnectionNodeIdToFile=1 or 0`	Legacy option for older Connection nodeversions.
`install.etxcn.CreateETXProxyUser=1 or 0`	When set to 1, creates the etxproxy user account if it does not exist.
`install.etxcn.CreateETXXstartUser=1 or 0`	When set to 1, creates the etxstart user account if it does not exist.
Service flags	
`install.service.createservice=1 or 0`	When set to 1, create service `otetxcn service`. It is recommended NOT to turn this off.
`install.service.bBootStart=1 or 0`	When set to 1, enables running the node when the system starts.
Node registration flags	

Property and Values	Description
`install.register.bAutoRegister=1 or 0`	When set to 1, the system attempts to automatically register the Connection Node at the end of the installation.
`install.register.r_serverurl=https://yourserver.fqdn:port`	Specifies the URL to connect to for registration.
`install.register.r_overridehostname=fqdn.hostname`	Specifies the hostname for the Exceed TurboX site. This is the value displayed by the Server Manager as **Host Name**.
`install.register.r_proxy=1 or 0`	When set to 1, enables the Proxy Manager role.
`install.register.r_auth=1 or 0`	When set to 1, enables the Authentication role.
`install.register.r_appscan=1 or 0`	When set to 1, enables application scanning.
`install.register.r_firstdisplay= display=display`	The starting display number for the X server. Must be a positive integer (1 or greater).
`install.register.r_altnameserver= fqdn.hostname`	A fully qualified hostname to use to establish connections from the server to this host.
`install. register.r_altnameclient= fqdn.hostname`	A fully qualified hostname to use to establish connections from the clients to this host.
`install. register.r_additionaloptions`	Used for undocumented flags or recommended override values.
`install.register.r_resttoken= api_ key`	The API key to use for REST authentication when registering the node. This API key must have been created by a user with full administrator rights.
`install.register.r_maxtotalsessions=int`	The maximum number of active and suspended sessions that can run on this node at one time.
	The value must be either 0, indicating no limit, or a positive integer.

Property and Values	Description
`install.register.r_maxsessp eruser= int`	The maximum number of active and suspended sessions that any one user can have on this node at one time.
	The value must be either 0, indicating no limit, or a positive integer.
`install.register.r_notes=st ring`	Enter a string that goes into the node information notes field. Do not include UNIX symbols such as '" * & $. If you want to include those symbols in a note, you can modify the note after installation using the Server Manager or the REST API.
`install.register.r_WebAdapt or=int`	Indicates whether to allow Web Client access on this node. Set to 1 to allow Web Client access or 0 not to.
`install.register.WebAdaptor Port=int`	When install.register.r_WebAdaptor is set to 1, set this to the port on which the Web Adaptor will listen for Web Client requests.
Installation flags	
`install.register.r_kerberos lib=/path/to/libkerb5`	This parameter overrides the default path toKerberos libraries, as detected by theinstaller. In most cases, the default path willbe correct and this parameter may be omitted. (Applicable to UNIX systems only.)
`install.register.r_gsslib=/ path/to/libgssapi`	This parameter overrides the default path to Kerberos libraries, as detected by the installer. In most cases, the default path will be correct and this parameter may be omitted. (Applicable to UNIX systems only.)

For additional tasks you can perform, see the following topics:

- Troubleshooting installation errors
- Registering a node
- Changing the node certificate

Troubleshooting installation errors

If the installation was successful, the `bin/install` script returns `0` (success). If the script returns a number other than `0`, a failure has occurred. A failure may be the result of a bad response file or may be caused by other issues, such as user permissions.

If the installation fails, examine the installation logs to determine the case. Installation log files are saved to the `/installlogs` directory in the installer root folder.

Registering a node

When using PAM or Native authentication, you must install and register an authentication node before users can sign in to Exceed TurboX Server. The Exceed TurboX Server administrator can always sign in as the `etxadmin` user, to configure authentication and register or configure the authentication nodes.

When the node registration was successful, the node state is set to **Enabled** in the Server Manager.

To automatically register the node during an installation (silent or not), Exceed TurboX Server must be running and able to accept REST API requests and authenticate users. You cannot silently install the first native authentication Connection Node with automatic registration.

If you did not register the node automatically during the node installation, you can perform this step at a later time, provided that Exceed TurboX Server is running and able to accept REST API requests. During the registration process you can do one of the following:

- Insert a token in the response file.
- Set the environment variable with the token.

Changing the node certificate

A self-signed SSL certificate is created during a silent installation. To use a custom certificate, either the installation must be migrated from a previous version or you can use the following steps.

To import an existing certificate:

1. Unpack the Connection Node into place.
2. Create customized `etxcert.pem` (public key) and `etxpriv.pem` (private key) files in `runtime/keys` (or links with `etxpriv.pem/etxcert.pem` naming).
3. Run the silent installation (it automatically skips the certificate generation and uses the customized `etxpriv.pem/etxcert.pem` files).

   ```
   bin/install -s fullpath/responsefile
   ```

Exceed TurboX Node silent installation on Windows systems

This section provides information about Exceed TurboX Node silent installation on Windows systems.

The installation CD for the Windows connection node includes the `setup.exe` file, which is an InstallShield package. You can pass MSI parameters as arguments to `setup.exe` to perform a silent installation of the Exceed TurboX node.

To install a node silently, run the following command:

```
setup.exe /v"/q UI_AUTOREGISTER=int UI_SERVERURL=url:port
UI_SERVERRESTTOKEN=token_string UI_REGISTER_PROXYROLE=int
UI_REGISTER_HOSTNAME=overridehostname.fqdn
UI_REGISTER_ALTSERVERNAME=altname.fqdn
UI_REGISTER_ALLOWNEWSESS=int
UI_REGISTER_MAXTOTALSESS=int
UI_REGISTER_MAXPERUSERSESS=int
UI_REGISTER_SSHCOMMAND=path_to_sshclient
UI_REGISTER_ALTCLIENTNAME=altname.fqdn UI_REGISTER_APPSCAN=int
UI_REGISTER_FIRSTDISPLAY=int"
```

You can configure the following MSI properties for a silent node installation on Windows.

Property and Values	Description
`UI_PORTNUMBER=(int)`	The port on which the Connection Node will listen for Exceed TurboX Server requests.
Silent installation configuration	
`UI_AUTOREGISTER=int`	When set to 1, the system attempts to automatically register the Connection Node at the end of the installation.
`UI_SERVERURL=url:port`	The URL to connect to for registration.
`UI_REGISTER_ALLOWNEWSESS=int`	Whether users may launch new sessions on this node. The default value is 0, indicating that new sessions may not be launched on this node.
`UI_REGISTER_MAXTOTALSESS=int`	The maximum number of active and suspended sessions that can run on this node at one time
	The value must be either 0, indicating no limit, or a positive integer.

Property and Values	Description
`UI_REGISTER_MAXPERUSERSESS=int`	The maximum number of active and suspended sessions that any single user can own on this node at one time.
	The value must be either 0, indicating no limit, or a positive integer.
`UI_REGISTER_SSHCOMMAND=path_to_sshclient`	Overrides the default path to the shell. In most cases, the path should not be changed from the default, and this parameter should be omitted. This parameter can be changed if you want to use a custom shell for launching Exceed TurboX profiles, or to change the path to the shell executable.
	Save the Secure Shell client executable file on the server before you specify the path of the executable in this property.
`UI_WEBADAPTOR=int`	Indicates whether to allow Web Client access on this node. Set to `1` to allow Web Client access or `0` not to.
`UI_ADAPTORPORTNUMBER=int`	When `UI_WEBADAPTOR` is set to `1`, set this to the port on which the Web Adaptor will listen for Web Client requests.
Authentication configuration	
`UI_SERVERRESTUSER=username` and `UI_SERVERRESTPASSWORD=cleartext!`	The user name and password to use to make REST API calls when registering the node.
	Note: It is recommended to use tokens and avoid using clear text passwords.
`UI_SERVERRESTTOKEN=token_string`	The API key to use for REST authentication when registering the node. This API key must have been created by a user with full administrator rights.
`UI_REGISTER_PROXYROLE=int`	When set to 1, this Connection Node will have the Proxy Manager role enabled.
`UI_REGISTER_HOSTNAME=overridehostname.fqdn`	The hostname for the Exceed TurboX site. This is the value displayed by the Server Manager as **Host Name**.
`UI_REGISTER_ALTSERVERNAME=altname.fqdn`	A fully qualified hostname that will be used to establish connections from the server to this host.

Property and Values	Description
`UI_REGISTER_ALTCLIENTNAME=altname.fqdn`	A fully qualified hostname that will be used to establish connections from the clients to this host.
`UI_REGISTER_APPSCAN=int`	When set to 1, application scanning will be enabled on this Connection Node.
`UI_REGISTER_FIRSTDISPLAY=int`	The starting display number for the X server. Must be a positive integer (1 or greater).

In addition, you can configure the following MSI property to suppress the automatic reboot process.

Microsoft-specific MSI Property	Description
`REBOOT=reallysuppress`	When this argument is set in the `/v" parameter=val"` statement along with other silent node installation properties, the automatic REBOOT process is disabled.

Automating the License Server registration

Exceed TurboX administrators can automate the Exceed TurboX installation and configuration process by using third-party tools and scripts that run the `bin/etxsvr config` command with specific values. This overrides the default values defined in the Server and allows administrators to customize the Server settings before starting the Server. For information about the Server settings you can configure, see Configuring Exceed TurboX Server settings.

Administrators can also automate the License Server request by using the following REST APIs provided with the product:

- `POST /v2/licenseserver/hostname` : Use this API to register `hostname` with the License Server.
- `DELETE /v2/licenseserver/hostname` : Use this API to unregister `hostname` with the License Server.
- `POST /v2/licenseserver/hostname/allocate/number` : Use this API to allocate `number` licenses to `hostname`. `'0'` indicate dynamic license allocation; other positive numbers indicate a static license allocation.

These APIs are called against a License Server system, not against a Server (which is a license consumer). For more information about REST APIs, see *Getting started with Exceed TurboX REST APIs* in the *Exceed TurboX Server Manager Help*.

Configuring web security settings

Configuring web security settings

This section describes how to modify web security settings in the `server_root/data/conf/override-web.xml` file. If this file does not exist, you may create it. A sample `override-web.xml` file is located in `server_root/lib/webapp/WEB-INF/config/override-web.xml.sample` for new installations of Exceed TurboX Server.

> **Note**
>
> The `override-web.xml` file must be created for each Exceed TurboX Server in a High Availability Server cluster. This file will not replicate automatically.

Content-Security-Policy header

The Content-Security-Policy HTTP header determines how resources, such as JavaScript and CSS, can be loaded by the browser. This header can protect users against clickjacking, Cross-Site Scripting (XSS), and malware injection. You can either enable Exceed TurboX Server's default Content-Security-Policy (CSP) or set a custom CSP in `override-web.xml`.

Example 1: Enable the default CSP in Exceed TurboX Server, which sets the following parameters (in addition to other internal flags):

default-src 'self' etx12: ; img-src 'self' data: ; script-src 'self' ; style-src 'self'

> **Note**
>
> Enabling the default CSP will prevent custom websites on other hosts from displaying the Exceed TurboX Server user interface. The default CSP is not enabled by default to prevent breaking customer environments where the Exceed TurboX Server pages are embedded. If you are not embedding Exceed TurboX Server in other sites then it is recommended that you enable the default CSP on all Exceed TurboX Servers.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
            http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
            id="ETX-OVERRIDE" version="3.0">
    <context-param>
        <param-name>enableCSP</param-name>
        <param-value>true</param-value>
    </context-param>
</web-app>
```

Example 2: Customize the default CSP. This will enable the default CSP and customize only the specified policies.

Note

You cannot override the *script-src* and *style-src* policies using this method.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
             http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
             id="ETX-OVERRIDE" version="3.0">
    <param-name>Content-Security-Policy</param-name>
        <param-value>
            default-src ‘self' https://trusted.url;
            frame-ancestors'self' https://trusted.url;
            connect-src'self' wss://trusted.url;
    </param-value>
</web-app>
```

Example 3: Replace the default CSP with the definition below. Using this method you can replace the *script-src* and *style-src* policies with less-secure policies, such as *'unsafe-inline'* and *'unsafe-hashes'*.

Caution

It is not recommended to replace the built-in policies with the less-secure policies.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns=http://java.sun.com/xml/ns/javaee
            xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
            xsi:schemaLocation=http://java.sun.com/xml/ns/javaee
            http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd
            id="ETX-OVERRIDE" version="3.0">
    <context-param>
        <param-name>disableBuiltinCSP</param-name>
        <param-value>true</param-value>
     </context-param>
    <param-name>Content-Security-Policy</param-name>
        <param-value>
            default-src'self' https://trusted.url;
            frame-ancestors'self' https://trusted.url;
            script-src'nonce-*'  https://trusted.url;
            style-src'nonce-*' https://trusted.url;
            connect-src'self' wss://trusted.url;
        </param-value>
```

Cross-origin header

Cross-Origin Resource Sharing (CORS) is an HTTP-header based mechanism that allows a server to indicate any other origins (domain, scheme, or port) than its own from which a browser should permit loading of resources. By default, Exceed TurboX Server will not permit hosts from other domains to invoke REST APIs. If you are planning to deploy a web server on a different domain from Exceed TurboX Server, and want that host to access Exceed TurboX Server APIs, you must allow access to the APIs from that domain in the `override-web.xml` file.

Example: Set the cross-origin (CORS) filter in override-web.xml to allow access from another domain

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://
java.sun.com/
             xml/ns/javaee/web-app_3_0.xsd"
             id="ETX-OVERRIDE" version="3.0">
    <filter>
        <filter-name>cross-origin</filter-name>
        <filter-class>org.eclipse.jetty.servlets.CrossOriginFilter</filter-class>
        <async-supported>true</async-supported>
        <init-param>
            <param-name>allowedOrigins</param-name>
            <param-value>*.trusted.domain</param-value>
        </init-param>
        <init-param>
            <param-name>allowedMethods</param-name>
            <param-value>GET,POST,HEAD</param-value>
        </init-param>
        <init-param>
            <param-name>allowedHeaders</param-name>
            <param-value>*</param-value>
        </init-param>
    </filter>
    <filter-mapping>
        <filter-name>cross-origin</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
</web-app>
```

4. Configuring the Exceed TurboX site

Configuring the Exceed TurboX site

This section describes how to set up the Exceed TurboX Site and Installation Nodes, as well as how to configure Exceed TurboX for initial use, and migrate an existing Exceed TurboX environment to the latest version.

This section includes the following topics:

Setting up the Exceed TurboX site

Setting up the Exceed TurboX site

This section describes how to set up an Exceed TurboX site after installing Exceed TurboX Server in your environment.

To set up an Exceed TurboX site, perform the following tasks:

1. Configure your environment to support the authentication method you plan to use. See authentication methods for more information.

   > **Note**
   >
   > Because XStart uses PAM with all authentication methods, it is recommended to have PAM configured on Exceed TurboX nodes.

2. If you plan to implement single sign-on (SSO) authentication based on Kerberos, configure your environment to support this functionality. See Configuring your environment for single sign-on authentication based on Kerberos for more information.

3. Sign in to Exceed TurboX Server Manager and configure the site settings, as necessary. See *Managing the site settings* in *Exceed TurboX Server Manager Help* for more information.

Authentication Methods

Exceed TurboX supports the following authentication methods:

Kerberos single sign-on

The single sign-on (SSO) authentication method, based on Kerberos, ensures secure communication by assigning a ticket to each user who signs in to the network. The ticket is then embedded in messages to identify the sender of the message.

For more information about configuring Kerberos, see *Configuring your environment for single sign-on authentication based on Kerberos* in *Exceed TurboX Installation and Configuration Guide*.

Native

When using the Native authentication method, Exceed TurboX users are authenticated based on the user credentials defined on the Linux and Windows operating system where the Exceed TurboX Connection Node is installed. The user accounts that use this authentication method must be created in your Linux or Windows environment.

PAM

The PAM authentication method ensures secure communication by passing user Client credentials to a pluggable authentication module (PAM) service, for verification. A PAM service must be installed and configured in your Linux or UNIX environment.

The PAM interface works with multiple methods, such as LDAP, 2FA, or Kerberos.

The PAM service name for Exceed TurboX Nodes is `exceed-connection-node`. In a Linux environment, create the Exceed TurboX PAM service file, `exceed-connection-node`, which typically resides at `/etc/pam.d/`.

The following example shows potential `/etc/pam.d/exceed-connection-node` configuration in a RedHat Linux environment.

```
#%PAM-1.0

auth        *required   pam_nologin.so

auth        include     password-auth

account     include     password-auth

session     required    pam_loginuid.so
```

```
session     include     password-auth

password    include     password-auth
```

File content must be customized based on your environment and PAM configuration.

LDAP

The LDAP authentication method ensures secure communication by authenticating users based on credentials supplied by an LDAP server. Exceed TurboX can retrieve user and group information from RFC2307-compliant LDAP servers. An LDAP server must be configured with the user accounts and groups you plan to use in Exceed TurboX.

OTDS

The OTDS (OpenText Directory Services) method allows Exceed TurboX to forward authentication requests to an OTDS server on your network. An OTDS server must be configured with the user accounts and groups you plan to use in Exceed TurboX.

OTDS is a stand-alone server that provides scalable authentication and user management services to enterprise applications.

OTDS supports a variety of authentication systems, including LDAP, Active Directory, SAML (Security Assertion Markup Language), OAuth 1.0/2.0, REST, web services auth, HTTP auth, and OpenID. It supports 2-factor authentication (2FA) for SAML/OAuth, as well as the ability to synchronize users across multiple directories and define which users can access which applications.

OTDS is available on Open Text My Support for all customers, free of charge.

For detailed information about installing, configuring, and administering OpenText Directory Services, see *OTDS Installation and Administration Guide* on Open Text My Support.

WAM

The WAM method forwards authentication requests to a Web Access Management solution, such as CA SiteMinder, which uses HTTP cookies to identify authenticated users.

Exceed TurboX Server identifies users by sign-in name. If you switch between authentication modes, you must ensure there are no collisions and the same sign-in names belong to the same person. If you cannot ensure this, to avoid the risk of unauthorized access, we recommend that you delete the user accounts that are likely to collide.

Configuring your environment for single sign-on authentication based on Kerberos

Configuring your environment for single sign-on authentication based on Kerberos

This section describes the tasks you need to perform to implement single sign-on (SSO) authentication based on Kerberos.

To allow users to automatically sign in to Exceed TurboX using their Microsoft Active Directory credentials, you can set up the Kerberos network authentication service in your environment. SSO authentication with Kerberos can be used with Exceed TurboX Dashboard and Exceed TurboX Server Manager.

To configure your environment for SSO authentication based on Kerberos:

1. Verifying Windows Server requirements
2. Creating a service principal user account in the Active Directory
3. Creating a keytab file
4. Configuring the principal user account delegation setting
5. Modifying the configuration file for Exceed TurboX Server
6. Configuring the client Web browser for Kerberos
7. Sign in to Exceed TurboX Server Manager and configure single sign-on authentication (SSO) based on Kerberos.

   You must also configure your environment to support a secondary authentication method (Native user credentials, LDAP, OTDS, or PAM).

   When Exceed TurboX is configured to use SSO authentication based on Kerberos, Exceed TurboX first attempts to use a Kerberos ticket to authenticate users. If the SSO authentication process fails, Exceed TurboX then attempts to use the secondary authentication method you select.

   For more information about authentication methods, see "Configuring user authentication settings" in the Exceed TurboX Server Manager Help.
8. Verifying the client Web browser for Kerberos
9. (Optional) Modifying the configuration file for Exceed TurboX connection nodes

For systems configured to use single sign-on authentication based on Kerberos, which need to run forwarded tickets for SSH Xstarts, there are a few extra steps.

To configure your environment for Kerberos and ensure that the system is able to run forwarded tickets for SSH Xstarts:

1. Create a user account in the Active Directory for the host keytab. See Creating a service principal user account in the Active Directory.

   Ensure the user account is for a host keytab.

2. Run the `ktpass` command. See Step 2 of Creating a keytab file.

   Ensure you use `host` instead of `HTTP` in the `ktpass` command, as indicated in the following example:

   ```
   ktpass -princ host/FQDN_host_name@domain -mapuser user_account -pass user_account_password -ptype KRB5_NT_PRINCIPAL
   -out filename.keytab
   ```

3. Configure delegation setting. See Configuring the principal user account delegation setting.

4. Transfer the host keytab to the machine on which you want to use SSH Xstart, and run `ktutil`. See Modifying the configuration file for Exceed TurboX Server.

   Ensure you transfer the file to the appropriate machine.

5. Sign in to the Exceed TurboX system on which you configured Kerberos.

6. Create or modify an Xstart profile. See *Adding Xstarts to a Custom startup profile* in the *Exceed TurboX Dashboard and Client Help*.

   Ensure you select **Secure Shell** as the startup method.

   Specify the machine on which you transferred the host keytab as the **X Application host**.

   Select **Use session credentials**.

7. Use the Xstart profile to launch a session. See *Launching a new session from the Profiles pane* in the *Exceed TurboX Dashboard and Client Help*.

For more information about Kerberos architecture and terminology, see Kerberos terminology and architecture.

Verifying Windows Server requirements

Make sure that a supported version of Windows is installed in your environment and configured as a domain controller (that is, a computer that runs Microsoft Active Directory).

To implement single sign-on authentication based on Kerberos, a Key Distribution Center (KDC) server must be set up in your environment. Any Windows Server 2008, 2008 R2, 2012, 2012 R2, or 2016 domain controller can be used as a Kerberos KDC server for Kerberos-based client and host systems.

Creating a service principal user account in the Active Directory

You must create an Active Directory user account to use as the Kerberos service principal.

> **Note**
>
> The following procedure describes how to create a user account in Windows Server 2012. The steps may vary if you are using a different version. For more information, see the documentation for your Windows Server version.

To create a Kerberos service principal user account in Windows Server 2012:

1. In the Active Directory, expand the section that corresponds to the domain in which you intend to install Exceed TurboX Server.
2. Right-click the folder where you want to create the user account. On the context menu, point to **New**, and select **User**.
3. Enter the user account and password information as needed. Make sure that **Password Never Expires** is selected.
4. Click **Next**.
5. Review the account information and click **Finish** to create the account.

Creating a keytab file

This section describes how to create a Kerberos keytab file, to map the user account you created in the previous task to an Exceed TurboX Server.

To create a keytab file:

1. On the Active Directory host, open a command prompt window.
2. Run the following command to generate a keytab for the service principal:

   ```
   ktpass -princ HTTP/FQDN_host_name@domain -mapuser service_principal_user_account -pass
   service_principal_user_password - ptype
   KRB5_NT_PRINCIPAL -out filename.keytab.
   ```

   where `FQDN_host_name` is the fully qualified name of the Exceed TurboX Server host, `domain` is the domain in which the host is located (for example, `example.com`), `service_principal_user_account` is the name of the service principal user account created in the previous task, and `file_name` is the name that you want to assign to the keytab file.

   > **Note**
   >
   > Write down the value of the `-princ` parameter. You will need this value later in this procedure.

When the keytab file is generated, the following output is displayed in the command prompt window:

```
Targeting domain controller: domain_controllerSuccessfully mapped HTTP/host_name to user_accountKey created.
```

Configuring the principal user account's delegation setting

In the Active Directory, you must allow the service principal user account to be trusted for delegation of the Kerberos service.

To configure the delegation setting:

1. In the right pane of the Active Directory, right-click the name of the service principal user account that you created, and select **Properties**.
2. On the **Delegation** tab, select **Trust this user for delegation to any service (Kerberos only)**.
3. Click **OK**.

Configuring the principal user account delegation setting

In the Active Directory, you must allow the service principal user account to be trusted for delegation of the Kerberos service.

To configure the delegation setting:

1. In the right pane of the Active Directory, right-click the name of the service principal user account that you created, and select **Properties**.
2. On the **Delegation** tab, select **Trust this user for delegation to any service (Kerberos only)**.
3. Click **OK**.

Modifying the configuration file for Exceed TurboX Server

You need to modify the Kerberos configuration file (`krb5.conf`) on the Exceed TurboX Server host, so that the Server is informed of the Active Directory, then run `ktutil`.

To modify the configuration file:

1. Modify the `krb5.conf` file as needed, such as the realm and server address.
2. Transfer the `filename` keytab file to Exceed TurboX Server.
3. Use the `ktutil` command on the host to include/merge the `filename` keytab file into `/etc/krb5.keytab`:

   ```
   ktutil
   ```

   ```
   rkt /etc/krb5.keytab
   ```

> **Note**
>
> You may need to modify the `/etc/hosts` file on the Exceed TurboX Server to include the IP address, domain name, fully qualified domain name, and hostname of the Active Directory server.

```
rkt filename.keytab

list

wkt /etc/krb5.keytab

quit
```

> **Note**
>
> If `/etc/krb5.keytab` does not exist, a warning message appears. You can ignore this warning message.

Configuring client Web browsers for Kerberos

After creating the keytab file and making it available to Exceed TurboX Server, you must configure the client Web browsers in your environment for Kerberos. Each client Web browser must be configured to allow Exceed TurboX Dashboard and/or Exceed TurboX Server Manager to authenticate against the domain in which Exceed TurboX Server is located.

When a user or administrator uses a Web browser that is configured in this way, Kerberos tickets are available immediately.

> **Note**
>
> The steps may vary depending on your Web browser version. For more information, see the documentation for your Web browser.

Configuring FireFox, Internet Explorer, and Chrome Browsers for Kerberos

To configure Mozilla Firefox for Kerberos (Windows, Linux, Mac):

1. In the address bar of Firefox, type `about:config` to list the current configuration options.
2. In the **Search** box, type `negotiate` to show only options containing that word.
3. Double-click the `network.negotiate-auth.trusted-uris` entry.
4. In the **Enter string value** dialog box, enter the name of the domain to authenticate against (for example, `.example.com`).

5. Double-click the `network.negotiate-auth.delegation-uris` entry and specify the same domain that you entered in step 4.

To configure Internet Explorer for Kerberos (Windows only):

1. In Internet Explorer, select **Tools**, then **Internet Options**.
2. On the **Security** tab, select **Local intranet** and click **Sites**.
3. Make sure that the **Include all sites that bypass the proxy server** and **Include all local (intranet) sites not listed in other zones** check boxes are selected.
4. Click **Advanced**.
5. In the **Websites** list, add the name of the domain to authenticate against (for example, `example.com`), then click **Close**.

   > **Note**
   >
   > If you are using the fully qualified domain name as the Exceed TurboX URL, you must complete this step. You must also complete this step in Internet Explorer if you are using Microsoft Edge as the browser.

6. Click **OK** in the **Local intranet site** dialog box.
7. On the **Security** tab, select **Local intranet** and click **Custom Level**.
8. In the **Security Settings** dialog box, scroll to the **User Authentication** area.
9. Select **Automatic logon only in Intranet zone**, then click **OK**.
10. Select the **Advanced** tab.
11. Scroll to the **Security** area and select **Enable Integrated Windows Authentication**.

    !!! note If you receive the following message when launching Internet Explorer: “Your browser is not supported by Exceed TurboX Dashboard”, navigate to **Tools** > **Compatibility View Settings**, and clear the option for **Display intranet sites in Compatibility View**.
12. Use the command line to run the following command:

    `chrome.exe --auth-server-whitelist="*.domain.com" --auth-negotiate-delegate-whitelist="*.domain.com"`

    where `domain` is the name of the domain to authenticate against.

To configure Google Chrome for Kerberos (Linux only):

1. Follow the steps in the **Set Up Policies** section of the following Linux Quick Start article:

   http://dev.chromium.org/administrators/linux-quick-start

2. For the step in the article that prompts you to insert content into the file you created, insert the following content:

```
{
"AuthServerWhitelist": "*.domain.com",
"AuthNegotiateDelegateWhitelist": "*.domain.com",
}
```

   where `domain` is the name of the domain to authenticate against.

3. At a command prompt, run the following commands:

   - `defaults write com.google.Chrome AuthServerWhitelist"*.domain.com"`
   - `defaults write com.google.Chrome AuthNegotiateDelegateWhitelist "*.domain.com"`

     where `domain` is the name of the domain to authenticate against.

4. Currently, configuration is not required to use Kerberos on Safari.

Verifying the client Web browser for Kerberos

After you have configured the client Web browser for Kerberos and set up Exceed TurboX Server to use Kerberos, you should verify that the browser runs as expected.

To verify the client Web browser:

1. Log on to a Windows, Linux, or Mac machine using a domain account.

   > **Note**
   >
   > For machines that are part of the same domain as the account you are logged on with, `krbtgt` is obtained automatically and single sign-on using Kerberos authentication is successful. If the machine is not part of the domain or you do not log in with a domain account, run `kinit` to obtain `krbtgt` .``

2. Launch the browser you configured in Configuring the client Web browser for Kerberos.
3. Navigate to your system's Exceed TurboX URL.

   You should be signed in without providing credentials.

Modifying the configuration file for Exceed TurboX connection nodes

Optionally, you may need to modify the Kerberos configuration file on Exceed TurboX connection node machines, so that the connection node machine is informed of the Active Directory.

To modify the configuration file:

1. Modify the `krb5.conf` file as needed, such as the realm and server address.
2. If you have not already done so, install and register the connection node. For more information see Installing Exceed TurboX Connection Nodes.

Managing Exceed TurboX Connection Nodes

Managing Exceed TurboX Connection Nodes

This section describes how to manage your connection nodes. This section includes the following topics:

- Managing and uninstalling Exceed TurboX Connection Nodes on Windows systems
- Managing and uninstalling Exceed TurboX Connection Nodes on Linux and UNIX systems

Managing and uninstalling Exceed TurboX Connection Nodes on Windows systems

This section describes how to start, stop, and uninstall Exceed TurboX Connection Nodes on Windows platforms.

> **Note**
>
> This section refers to session nodes only. For information about starting, stopping, and uninstalling Connection Nodes in an HA cluster environment (Linux and Solaris platforms only), see *Exceed TurboX High Availability Configuration Guide*.

Windows-based Connection Nodes cannot provide PAM authentication like UNIX nodes, so you can uninstall a Windows node without affecting your users' ability to log on to the Exceed TurboX Site.

On Windows, the Connection Node runs as a service. You can use the Services application from your Windows Control Panel to see whether the OpenText ETX Connection Node service is running. You can also use your Windows Control Panel to start or stop the Connection Node service.

To uninstall a node:

1. Go to the **Apps & Features** or **Programs & Features** section of the Windows Control Panel.
2. At the **OpenText ETX Connection Node** entry, click **Uninstall**.

   The **Connection Node Install Wizard** is displayed.
3. Follow the prompts in the **Connection Node Install Wizard** to either repair or remove the Connection Node.

   Because the datastore has a unique entry for each node, the Install Wizard will unregister the node before removing all binaries and files.
4. Delete the node by invoking the `DELETE /v2/nodes/nodeid` REST API from Exceed TurboX Server.

> **Tip**
>
> You can also delete a node using the **Nodes and Sessions** tab in the Server Manager. For more information, see "Monitoring the nodes in the Nodes pane" in the *Exceed TurboX Server Manager Administration Guide*.

To re-register a node without uninstalling:

1. Stop all sessions running on the node that the user wants to re-register.

   To minimize disruption caused by stopping sessions, it is recommended that you disable the **Launch new sessions** option for the Node, and direct users to terminate sessions on the Node.
2. To unregister the node from the server, run the following command:

   `nodecmds unregister server_url`

   For example: `nodecmds unregister http://etxserver.example.com:8443`
3. To reinitialize the node's database, run the following command:

   `nodecmds reinitialize [new_port]`

   Specify the `new_port` value if a different node port is desired, else the same port number will be used. For example: `nodecmds reinitialize 5010`
4. To register the node on another server, run the following command:

   `nodecmds registernode new_port`

   For more information about registering a node on other server, see Registering a node.

Managing and uninstalling Exceed TurboX Connection Nodes on Linux and UNIX systems

This section describes how to start, stop, and uninstall Exceed TurboX Connection Nodes using the script provided in the product package. These scripts also provide commands for viewing node status.

> **Note**
>
> At least one node must be installed and registered with Exceed TurboX Server.

For node-maintenance purposes, the following scripts and binary are included in the Exceed TurboX Connection Node package:

Script/binary	Description
`bin/nodecmds`	This script is the main entry point for all maintenance tasks.

For a node to provide either the Authentication or Proxy Manager services to the Exceed TurboX site, it must be registered in the site datastore and the local `etxpm` binary must be started. These operations must be performed on each local node rather than in Server Manager.

The following tasks help maintain the node for proper operation.

To start a node when the system starts:

During node installation, administrators are asked whether they want the node to start automatically when the system starts. If enabled, the node will start when the system starts, and will add a systemd startup unit, if the system supports systemd. If the system does not support systemd, the node will install a SysV init script instead.

If bootstart was not enabled during node installation and you later want to enable it, run `otetxcn bootstart enable`.

Notes:

- Bootstart commands only determine whether the node will be started when the system is booted. They do not start or stop the node in the current system session.
- For a node to have access to the proper user environments, it must be started using bootstart. If you start a node using otexcn start, it will not have access to user environment data.
- Open a Terminal window.
- Run the `otetxcn bootstart enable` command. For example:

  `/opt/etxcn/bin/otetxcn bootstart enable`

To prevent a node from starting when the system starts:

To prevent a node from starting when the system starts, without uninstalling the node, run `otetxcn bootstart disable`.

1. Open a Terminal window.
2. Run the `otetxcn bootstart disable` command. For example:

   `/opt/etxcn/bin/otetxcn bootstart disable`

To check node running status:

Run `otetxcn status` to see the current running status of the node.

> **Note**
>
> If your Operating System uses systemd, you may get additional details by running `systemctl status otetxcn`.

1. Open a Terminal window.
2. Run the `otetxcn status` command. For example:

   `opt/etxcn/bin/otetxcn bootstart status`

   A node that shows the disconnected state has typically not been started or is no longer running.

To start a node:

> **Note**
>
> For a node to have access to the proper user environments, it must be started using `bootstart`, as described above. If you start a node using `otexcn start`, it will not have access to user environment data.

1. Open a Terminal window.
2. Run the `otetxcn start` command. For example:

   ```
   /opt/etxcn/bin/otetxcn start
   ```

   If this command is successful, the `etxpm` processes should be running.

To stop a node:

If you are not an Exceed TurboX administrator but you have administrative access to the node, you might need to stop the node to do maintenance on the server. This operation puts the node in a Disabled state and prevents the Site with which the node is registered from using it. The Exceed TurboX sessions running on the node will not be killed.

> **Note**
>
> If the node has the role of Proxy Manager, there may be running sessions on this machine. The following instructions will not affect any of these sessions. The node will be placed in a Disabled state, which means that users will not be able to resume, transfer, share or shadow any of these sessions.

If the node has the role of Authenticator, before you stop this node, ensure that another node providing Authentication is available. Otherwise the site will become unusable. Security tokens will expire and users will be logged out of the Dashboard and Server Manager.

1. Stop all sessions running on the node.

   For more information about stopping sessions, see *Terminating or suspending sessions* in the *Exceed TurboX Server Manager Administration Guide*.
2. Open a Terminal window.
3. Run the `otetxcn stop` command. For example:

   ```
   /opt/etxcn/bin/otetxcn stop
   ```

To remove a node:

To safely remove a Connection Node, the Node and all sessions running on it must be stopped. To minimize disruption caused by stopping sessions, it is recommended that you disable the **Launch new sessions** option for the Node and direct users to terminate sessions on the Node.

Because the Exceed TurboX Server datastore has a unique entry for each node, the node must be unregistered from Exceed TurboX Server before all binaries and files are removed. For information about stopping sessions before the node binaries and files are removed, see *Terminating or suspending sessions* in the *Exceed TurboX Server Manager Administration Guide*. For information about configuring node settings, see *Configuring node settings* in the *Exceed TurboX Server Manager Administration Guide*.

> **Note**
>
> This section refers to session nodes only. For information about starting, stopping, and uninstalling Connection Nodes in an HA cluster environment, see Exceed TurboX High Availability Configuration Guide.

1. Open a Terminal window.
2. To stop the node and sessions, and remove bootstart entries and files, run the `otetxcn remove` command. For example:

   ```
   /opt/etxcn/bin/otetxcn remove
   ```

   > **Note**
   >
   > To delete the files, you must execute the command from outside the node root directory. After you remove a node, the name of the node will be prepended with `z:id:hostname:port` in the Activity log. This is to permit the same host to be registered with the Site again, while ensuring that the two separate registrations remain distinct.

3. Delete the node by invoking the `DELETE /v2/nodes/{nodeid}` REST API from Exceed TurboX Server.

   > **Tip**
   >
   > Alternatively, you can delete the node using the **Nodes and Sessions** tab in the Server Manager. For more information, see *Monitoring the nodes in the Nodes pane* in the *Exceed TurboX Server Manager Administration Guide*.

Configuring Exceed TurboX for initial use

Configuring Exceed TurboX for initial use

This section describes tasks you need to complete after installing the product and before you make it available for use.

This section contains the following topics:

Mandatory post-installation configuration

This section describes the configurations you must perform after you install Exceed TurboX for the first time.

These include:

1. Create an Exceed TurboX administrator account.

   Sign in to Exceed TurboX Server Manager using the `etxadmin` account, and create a user with the **Admin (Full Access)** role. See *Signing in to Exceed TurboX Server Manager using the etxadmin account* and *Managing users* in the *Exceed TurboX Server Manager Help* for detailed information.

2. Sign in to Exceed TurboX Server Manager using an administrator account. See *Signing in to Exceed TurboX Server Manager* in the *Exceed TurboX Server Manager Help* for detailed information.

3. Apply the product license.

   In the Exceed TurboX Server Manager, go to the **Site Settings Licenses** page and apply the product license, as necessary. See *Managing licenses* in the *Exceed TurboX Server Manager Help* for detailed information.

4. Assign a role to the Exceed TurboX Connection Node.

   Go to the **Nodes and Sessions** page and use the **Nodes** pane to configure the nodes in your environment. See *Managing nodes* in the *Exceed TurboX Server Manager Help* for detailed information.

> **Note**
>
> In a multi-node environment (Linux and UNIX platforms only), if you are using Native or PAM authentication, at least one Connection Node must be designated as Authenticator. In all environments, in order to launch sessions, at least one Connection Node must be designated Proxy Manager.

Exceed TurboX is now ready for use with default settings. Additional configurations may be necessary, depending on administrative preferences. See Other post-installation considerations for more information.

Other post-installation considerations

Other post-installation considerations

This section describes several post-installation activities that you may consider before deploying the product to users.

> **Note**
>
> For descriptions of all available administrative configuration options, see the * Exceed TurboX Server Manager Help*.

This section includes the following topics:

- Configuring default permissions for new users
- Embedding the Exceed TurboX web interface in another webpage
- Enabling direct access to Exceed TurboX Dashboard using a login token
- Enabling printing for your users
- Using custom and scripts
- Adding a signed SSL certificate to Exceed TurboX

Configuring default permissions for new users

Use the **New User Settings** page of the Server Manager to configure default user settings, and to control user access to Exceed TurboX Dashboard features. The user permissions you define here are automatically applied to all new users. You can also apply the changes you make to the default permissions to all existing users, and to users who belong to user groups.

For detailed information, see *Configuring default user permissions and settings* in the *Exceed TurboX Server Manager Help*.

> **Note**
>
> You can configure permissions for *individual* users in the **Users and Profiles** page of the Server Manager. See *Modifying user settings, roles, and permissions for individual users* in the *Exceed TurboX Server Manager Help*. Individual settings override default settings.

Embedding the Exceed TurboX web interface in another webpage

This section describes how to configure the domains which are allowed to embed the Exceed TurboX web interface.

To prevent clickjacking attacks, the default configuration for Exceed TurboX prevents the Exceed TurboX web interface from being embedded in webpages on different domains. However, in certain cases (for example, if you want to embed Exceed TurboX into your company intranet) the Exceed TurboX Server can be configured to allow embedding, regardless of the domain.

To configure this feature:

1. In Exceed TurboX Server Manager, navigate to the **Site Settings** tab, **General** page.
2. In the **Optional settings** box, enter one of the following flags:
   - To allow any webpage (regardless of domain) to embed Exceed TurboX Server:

     `server.XFrameOptions=off`
   - To prevent all webpages from embedding Exceed TurboX Server:

     `server.XFrameOptions=DENY`
   - To allow only the same domain to embed Exceed TurboX Server:

     `server.XFrameOptions=SAMEORIGIN`

> **Note**
>
> If no `server.XFrameOptions` setting is entered, the `server.XFrameOptions=SAMEORIGIN` is assumed by default.

3. Click **Save**.

Web pages which embed Exceed TurboX Server can also pass login credentials to the embedded server page. This allows the parent site to log users in automatically, using credentials stored inside a login token. For more information, see Enabling direct access to Exceed TurboX Dashboard using a login token.

Enabling direct access to Exceed TurboX Dashboard using a login token

This section describes how to log users in to Exceed TurboX programmatically, using credentials stored inside a login token.

This feature allows administrators to integrate Exceed TurboX Dashboard into the company website and authorize users to access Exceed TurboX Dashboard automatically, without having to log in to Exceed TurboX separately (the Exceed TurboX Dashboard sign-in page does not appear). For example, users can log in to their corporate website and click a web link to access Exceed TurboX Dashboard without entering credentials again.

To enable direct access to Exceed TurboX Dashboard using a login token:

1. Using a custom server-side script, call the following REST API and pass the user's user name and password as parameters:

   `POST /site/util/encrypt`

   The call returns a login token that can be used to log the user in to Exceed TurboX Dashboard. Administrators may call the REST API with their preferred server-side programming technology, such as PHP, Perl, or JSP.

2. Create the "direct login" URL by combining the Exceed TurboX Dashboard URL with the login token generated in Step 1. The "direct login" URL uses the following format:

   `ETX_URL?login=token`

3. Dynamically insert this "direct login" URL into your website, so it appears as a link in the end user's web browser.

Example:

1. Run the following command:

   ```
   curl -X POST -k -u ETX admin username:ETX admin password --data username=username
   for automated login --data password=password for
   automated login https://etxs.example.com/etx/api/site/util/encrypt
   ```

   The REST API call returns a URL safe login token in a text/plain format. For security reasons, the token is valid for 60 seconds only.

2. Create a webpage that uses the acquired login token using the new `"?login=token"` parameter.

   ```
   https://etx server/etx/?
   login=57kiQoixB113X4uNWGH36GBwbI22FDsXjsgSabcNqDzTe5k0TKTd2lIDBFoZnxkkqoEUKhYsMLbXtoqoSO8hCwibmDEacsWgnrZxmnT
   ```

> **Note**
>
> Users that have never logged in to Exceed TurboX and do not have an Exceed TurboX user account may still log in with this method, if the **New users can sign in with successful authentication** setting is enabled in the Exceed TurboX site settings. In this case, the user's account will be created as normal after their first login.

Enabling printing for your users

Exceed TurboX allows you to print files from the command line using the Exceed TurboX File Transfer and Remote Print Utility (`etxft`), available in the product package in the `Utilities/etxft` directory.

Users who launch a session on a UNIX/Linux host can use a Common UNIX Printing System (CUPS) server to print files. Files are sent to the Exceed TurboX Connection Node using the Exceed TurboX File Transfer and Remote Print Utility (`etxft`), then transferred by the Connection Node to the end user's machine, where local printing occurs.

To use this feature, CUPS must be installed and running on your UNIX/Linux application host. For more information, see http://www.cups.org.

You must also install the CUPS printer by running the `cups` installation script from the Exceed TurboX installation package.

For detailed information, see "Transferring and printing files with Exceed TurboX" in the *Exceed TurboX Dashboard and Client Help*.

Using custom `pre-session` and `post-session` scripts

The Exceed TurboX node can run a custom script before a session starts as well as a separate script after the session terminates. There are often circumstances where additional actions are needed for smooth maintenance of the connection node. For example, when a session terminates on a node, you can restart the node to clean up old or unwanted processes.

The `pre-session` and `post-session` script files are supported only on the Unix platforms.

To run `pre-session` and `post-session` scripts:

1. Create a script named `pre-session` as `etxpm` and install the script in the `ETXCN_home/bin/` location.

   The script is executed by `etxpm` process before a session is created as follows:

   `ETXCN_ROOT/bin/pre-session`

2. Create a script named `post-session` as `etxpm` and install the script in the `ETXCN_home/bin/` location.

   The script is executed by `etxpm` process after the termination of a session including a failure as follows:

   `ETXCN_ROOT/bin/post-session`

> **Note**
>
> The `post-session` script must run only after receiving all communications about the termination of a session. Otherwise, the script can shut down the system immediately (and permanently) and can block further communication with the Server.

When the scripts are run, the following environment variables are available for the scripts to use:

- `ETX_USERNAME` = `username`
- `ETX_USERID` = `user ID`
- `ETX_DISPLAY_NUM` = `n`
- `ETX_PROFILE` = `profile name`
- `ETX_PROFILEID` = `profile ID`
- `ETX_SESSIONID` = `session ID`

The `pre-session` and `post-session` script files must meet the following criteria:

- Is owned by the `etxpm` installer (`ETXCN` home directory owner).

- Does not have the Write permission for any user other than owner.
- Is executable only by the file owner.

If any of the preceding criteria is not met, then `etxpm` logs an appropriate error message and ignores the script.

> **Note**
>
> The script file can be of any type (for example, bash, perl, and so on) or a binary file that can be executed from a shell.

Adding a signed SSL certificate to Exceed TurboX

Exceed TurboX Server versions 12 and later automatically generates a private key for this server. This is a standalone key, and is not trusted/produced by a Root Certificate Authority (CA). This type of key is considered insecure because anyone can make such a key. To get modern browsers to trust the site automatically, you need a certificate provided by a CA.

To generate a signed certificate:

1. Generate a private key for the Exceed TurboX Server for which you need the certificate.
2. Generate a Certificate Signing Request file (typically, CSR format) for the server for which you need the certificate, that uses the private key for that server. Be sure to provide all required information.

   Example:

   The following example is the standard method to produce a SHA256 RSA-based CSR. It results in a PEM-formated `your_server_fqdn.key` file.

```
 # generate CSR and private key together
 openssl req -x509 -nodes -sha256 -newkey rsa:2048 -keyout your_server_fqdn.key -
out your_server_fqdn.crt

 # we can add more parameters
 # interactive output
 writing new private key to 'your_server_fqdn.key'
 -----
 You are about to be asked to enter information that will be incorporated
 into your certificate request.
 What you are about to enter is what is called a Distinguished Name or a DN.
 There are quite a few fields but you can leave some blank
 For some fields there will be a default value,
 If you enter'.', the field will be left blank.
 -----
 Country Name (2 letter code) [US]:
 State or Province Name (full name) []:
 Locality Name (e.g., city) []:
 Organization Name (e.g., company) []: Your company name
 Organizational Unit Name (e.g., section) []: []
 Common Name (e.g., your name or your server's hostname) []:your_server_fqdn
 Email Address []: your e-mail
```

3. Provide the CSR to the CA website, usually by uploading or by copying/pasting into a field.
4. Download the public certificate file from the CA.
5. Convert the private key and public certificate file to be used by the Exceed TurboX Server to PEM format. Exceed TurboX Server version 12 (and later) expects both the `server.crt` and `server.key` to be in PEM format.

   The `server.crt` file should have the entire CHAIN of certificates from the Root Central Authority (CA) all the way down to the local server certificate. Typically, there are Intermediate certificates in between the ROOT and the SERVER. The PEM file can have many HEADER/CERT/FOOTER elements concatenated together.

```
-----BEGIN CERTIFICATE-----
 ( server specific certificate here)
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
(Your Intermediate certificates)
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
(Your Root certificate here)
-----END CERTIFICATE-----
```

> **Note**
>
> When you are using Firefox, the client needs the root CA in the local browser, otherwise the web browser shows only the server certificate, instead of the whole chain of certificates.

The `server.key` should only have the private key information that goes with the Server.

You need to ensure that your PEM file is in the PKCS#8 format for the private key. For example, if your PEM file is in the PKCS#1 format as follows:

```
-----BEGIN RSA PRIVATE KEY-----
(private key data)
-----END RSA PRIVATE KEY-----
```

Convert it to the PKCS#8 format as follows:

```
-----BEGIN PRIVATE KEY-----
( private key base64 DER encoded X.509 here)
-----END PRIVATE KEY-----
```

What you need to do in this step depends on the format of the public certificate file you downloaded from the CA in Step 4 above. If you downloaded a .P12 or .PFX file, you need to construct the `server.key` using one of the following commands:

```
openssl pkcs12 -in file_from_CA.p12 -clcerts -out serveronly.crt
openssl pkcs12 -in file_from_CA.p12 -cacerts -out server-ca.crt

<< a CHAIN of certificates needs to follow as described in step 5 above >>

cat serveronly.crt server-ca.crt > server.crt
```

6. Review the final `server.key` and `server.crt` files. Ensure that they include only `CERTIFICATE` or `PRIVATE KEY` lines, otherwise the keys will not work correctly with Exceed TurboX. Remove any lines from the beginning of the CRT file, which are not necessary. For example, remove all `Bag Attributes` preceding the `-----BEGIN CERTIFICATE-----`, as shown in the following example.

```
Bag Attributes
    localKeyID: 01 00 00 00
    1.3.6.1.4.1.311.17.3.20: C4 60 A5 7F B2 37 C0 ....
    friendlyName: somefriendlynamehere
    1.3.6.1.4.1.311.17.3.13: 6A 00 2D 00 6C 00 65 00 ....
1.3.6.1.4.1.311.17.3.71: 57 00 49 00 41 00 44 00 4D ....
    1.3.6.1.4.1.311.17.3.87: 00 00 00 00 00 00 00 00 ....
subject=/CN=somethinghere
issuer=/DC=ca/somethingelsehere
-----BEGIN CERTIFICATE-----
<cert data>
-----END CERTIFICATE-----
```

7. Install the Private key. Copy the file as `server_root/data/ssl/server.key`.
8. Install the Public key. Copy the files as `server_root/data/ssl/server.crt`.

Migrating to the latest version of Exceed TurboX

Exceed TurboX 12.5.x supports migration only from earlier 12.0.x versions. If you are using Exceed TurboX 11.x, please follow the instructions in the *Installation and Configuration Guide* for Exceed TurboX 12.0.3 to migrate to that version. Once that migration is finished and you have made sure it was successful, follow the instructions in Applying Exceed TurboX service pack updates to install version 12.5.x as a service pack.

5. Applying Exceed TurboX service pack updates

Applying Exceed TurboX service pack updates

This section describes how to apply updates released in Exceed TurboX service packs (SPs).

Exceed TurboX service packs provide new features, enhancements, and bug fixes for the following product areas:

- Exceed TurboX Server
- Exceed TurboX Connection Nodes
- Exceed TurboX Client Launcher
- Exceed TurboX Runtime

It is recommended that you apply service packs during off-hours to avoid network congestion and the possibility of delays when launching new sessions.

For instructions on applying service packs in an Exceed TurboX Server high-availability cluster environment, see *Exceed TurboX High Availability Configuration Guide*.

> **Note**
>
> To perform a full installation (or a re-installation) of Exceed TurboX instead of patching your existing Exceed TurboX environment, see Installing Exceed TurboX Server components.

To apply a service pack to an existing Exceed TurboX site:

1. Download the service pack and extract the service pack components (see Downloading the service pack).
2. Apply the service pack to Exceed TurboX Server (see Applying a service pack update to Exceed TurboX Server).
3. If a new runtime is provided, install the runtime (see Installing a runtime using automated scripts).
4. If necessary, apply any updates related to keyboards, fonts, and third-party components included in the service pack (see Applying other service pack updates).

> **Note**
>
> Starting in Exceed TurboX 12.0, the Exceed TurboX Launcher silently updates itself to the latest Launcher version, which is embedded in the Exceed TurboX Server. However, the 12.0 Launcher cannot self-update to support Kerberos authentication. If you are currently using an older Launcher version that does not support Kerberos, and you want to enable Kerberos authentication, manually uninstall the 12.0 Launcher and re-install the 12.x Launcher.

To update the Launcher manually, download the latest version from the **User settings** dialog box on the Exceed TurboX web UI. For more information, see *Downloading and installing the Client Launcher* in the *Exceed TurboX Dashboard and Client Help*, or in the *Exceed TurboX Server Manager Help*.

In Exceed TurboX, you can use the launcher automatic upgrade capability for Windows and Linux platforms if you are using an authentication method other than Kerberos.

Downloading the service pack

This section describes how to download the service pack for your Exceed TurboX version.

To download the service pack:

1. Download the service pack (`.ZIP` file) for your Exceed TurboX version.

   If you have issues downloading the service pack, contact customer support.
2. Extract the downloaded file.

   The contents of the directory are displayed.

   The contents of the directory vary with product updates. The following is a list of common contents:

   - `ETXServer`
   - `ETXRuntime` (optional)

     > **Note**
     >
     > Runtime packages are provided separately for customers who want to update the runtime without updating Exceed TurboX Server.

   - `ETXLauncher`

> **Note**
>
> You can also download Launcher packages, including the Windows per-machine package, from the **User properties** dialog box in Server Manager. For more information, see *Downloading and installing the Client Launcher* in the Exceed TurboX Server Manager Help.

- `ETXConnectionNode`

> **Note**
>
> Exceed TurboX Connection Node files are packaged inside some Exceed TurboX Server patches. Applying an update to Exceed TurboX Server automatically updates all Connection Nodes to the same version as Exceed TurboX Server. The automated patching process eliminates the need to manually patch each node and ensures that the Connection Nodes always use the same version as Exceed TurboX Server.

> **Note**
>
> The Exceed TurboX Connection Node package also includes files that are not updated by the Server automatic patching process. These files are VirtualGL (for Linux), keyboard files, font files, and the Space Ball utility. Customers who want these updated files must extract the Connection Node patches over the patched nodes. Exceed TurboX includes updated VirtualGL (for Linux nodes), as well as updated keyboard and font files (for all node platforms).

Applying a service pack update to Exceed TurboX Server

Applying a service pack update to Exceed TurboX Server

This section describes how to apply the server pack update to Exceed TurboX Server on supported Linux, Solaris, and Windows platforms:

- Applying a service pack update on Linux and Solaris platforms
- Applying a hotfix update on Windows platforms

Notes

- Patching must be done by the same user that installed the Server (that is, the user that owns the `data/etxdata` directory).
- On Windows platforms, service pack updates follow the same procedure as Server installation. See Installing Exceed TurboX Server on a Windows platform for more information.

Applying a service pack update on Linux and Solaris platforms

This section describes how to apply the server pack update to Exceed TurboX Server on supported Linux and Solaris platforms.

The Server pack file uses the following naming convention:

```
ETXServer-ETX Version.Build Number-SP Number-Platform-CPU.sh
```

For example:

```
ETXServer-12.0.0.5790-SP1-linux-x64.sh
```

> **Note**
>
> Patching must be done by the same user that installed the Server (that is, the user that owns the `data/etxdata` directory).

In the following procedure, file and folder names may be for a different version than the version you are currently installing. Use the correct paths and executable names for the version you are applying.

To apply the server pack update to Exceed TurboX Server (Linux platforms):

1. Back up the entire Exceed TurboX Server directory.

> **Important**
>
> The script overwrites all files in the Server directory with the new files provided in the patch. Ensure you back up your original files before applying this patch.

2. Copy the `.sh` file for your platform into the Exceed TurboX Server directory.
3. Change directory to the root of the installation.

```
cd /opt/etxsvr/bin/etxsvr-<version>
```

4. Run one of the following commands to stop the Exceed TurboX Server:

   - If the Server is running as an application:

`./bin/etxsvr stop`

- If the Server is running as a service:

  `systemctl stop otetxsvr`

  or

  `systemctl stop otetxsvr_n`

5. If you are applying a 12.5.x service patch to an earlier version of Exceed TurboX Server, disable the bootstart. Otherwise, proceed to step 6.

   `bin/etxsvr bootstart disable`

6. Run the update script.

   For example:

   `sh ./ETXServer-12.0.0.5790-SP1-linux-x64.sh`

   The script runs and determines if the contents of the patch must be applied to this Exceed TurboX Server.

7. If any content is applicable, type `y` or press enter to apply the patch.

   The script runs, applying the update.

8. If you are applying a 12.5.x service patch to an earlier version of Exceed TurboX Server, perform the following steps. Otherwise, proceed to step 8.

   a. Change directory to the Exceed TurboX Server Installation folder:

   `cd /opt/etxsvr/bin/etxsvr-<version>`

   b. Enter the following command to create the systemd scripts:

   `bin/etxsvr bootstart enable`

   The command prints the name of the systemd service it created. For example, `otetxsvr`.

   If there is more than one instance of etxsvr on the computer, the name of the service has the format `otetxsvr_n`, where `n` is a unique identifying number. For example, `otetxsvr_2`.

9. Enter one of the following commands to restart the server:

   - If the Server is running as an application:

     `./bin/etxsvr start`

   - If the Server is running as a service:

     `systemctl start otetxsvr`

     or

     `systemctl start otetxsvr_n`

The Exceed TurboX Server patch is now applied.

The Server and the Connection Nodes are now running the latest version.

> **Note**
>
> The latest version of the Client Launcher is available for download from the Exceed TurboX web UI.

10. To verify if Exceed TurboX Server is running the new service pack version, enter the following command:

```
./bin/etxsvr version
```

Applying a hotfix update on Windows platforms

This section describes how to apply a hotfix update to Exceed TurboX Server on supported Windows platforms.

The Server pack file uses the following naming convention:

```
ETXServer-ETX Version.Build Number-SP Number-Platform-CPU.zip
```

For example:

```
ETXServer-12.5.0.7991-SP1-windows-x64.zip
```

> **Note**
>
> Do not rename the `.ZIP` file. Renaming the file will prevent the `etxpatch` application from applying the patch.

In the following procedure, file and folder names may be for a different version than the version you are applying. Use the correct paths and file names for the version you are applying.

This procedure is for hotfix updates only. Service pack updates follow the same procedure as Server installation. See Installing Exceed TurboX Server on a Windows platform for more information.

To apply the hotfix update to Exceed TurboX Server on Windows platforms:

1. Click on the **Administrative Command Prompt for ETX Server** shortcut.
2. When you are prompted to elevate, click **Yes**.

   The **Administrator** command prompt window opens.

3. Back up the entire Exceed TurboX Server directory.

> **Important**
>
> The patch application overwrites all files in the Server directory with the new files provided in the patch. Ensure you back up your original files before applying this patch.

4. Copy the `.ZIP` file into the patch folder.

   `C:\Program File\OpenText\ETXServer\patch`

5. Change directory to the root of the installation.

   `cd C:\Program Files\OpenText\ETXServer`

6. Stop Exceed TurboX Server.

   `.\bin\etxsvr stop`

7. Run the update script.

   `.\patch\bin\etxpatch.exe`

   The `etxpatch` application runs and determines if the contents of the patch must be applied to the Exceed TurboX Server. The application applies the update, if applicable.

   The `Bin\etxpatch.exe` binary generates the `logs\etxpatch.log` log file.

   Once the patch is complete, the `.ZIP` file is deleted automatically.

8. Once the patch is complete, restart the server.

   `.\bin\etxsvr start`

   The Exceed TurboX Server patch is now applied. The Server and Connection Nodes are now running the latest version.

   > **Note**
   >
   > The latest version of the Client Launcher is available for download from the Exceed TurboX web UI.

9. To verify that Exceed TurboX Server is running the new version, type the following command:

   `.\bin\etxsvr version`

Installing a runtime using automated scripts

This section provides instructions for installing the runtime using automated scripts in Linux/UNIX and Windows environments.

> **Note**
>
> You can also install a runtime using the **Runtimes** tab in Server Manager. See *Exceed TurboX Server Manager Help* for more information.

Exceed TurboX Client and Proxy updates are issued as runtime (`.ZIP`) files (Linux, UNIX, and Windows platforms), and are included in `.msi` files (Windows platforms only). Exceed TurboX Server can support multiple runtime versions, one of which must be designated as the default.

Applying Exceed TurboX Server patches, such as service packs and major/minor version patch scripts (`.ZIP` on Windows; `.sh` on Linux/UNIX), does not automatically install a new runtime. This is because runtimes are versioned independently from Server patches. After you apply the service pack update, you must extract and add the runtime separately. Once you've extracted the runtime, you can designate it as the default runtime. (The default runtime is used when a profile does not have a specific runtime associated with it.)

On Windows platforms, if you are installing an update from an `.msi` file, you do not need to install the runtime separately. If you are installing a hotfix from a `.ZIP` file, you are required to install the runtime separately. For information about installing an update using an `.msi` file, see Installing Exceed TurboX Server on a Windows platform.

To extract the runtime:

1. Extract the runtime `.ZIP` file (`etxsvr_root/data/runtimes/patch/runtime_version`) by running the `mkdir -p` command.

   For example:

   - Linux/UNIX: `mkdir -p /opt/etxsvr/data/runtimes/patch/12.0.2.6288`
   - Windows: `mkdir C:\Program Files\OpenText\ETXServer\data\runtimes\patch\12.5.0.8000`

   > **Note**
   >
   > Ensure the version of the runtime you extract matches the folder you extract it to.

2. Unzip the runtime.

To make the extracted runtime the default runtime: (optional)

Create a file with a specific name in the new runtime folder:

```
<etxsvr_root>/data/runtimes/patch/<runtime_version>/makeDefault
```

For example:

- Linux/UNIX:

```
touch /opt/etxsvr/data/runtimes/patch/12.0.2.6288/makeDefault
```

- Windows:

```
C:\Program Files\OpenText\ETXServer\data\runtimes\patch\12.5.0.1\makeDefault
```

To restart the Server to pick up the runtime:

Restart the Exceed TurboX Server using the following command:

```
etxsvr_root/bin/etxsvr restart
```

To add a runtime

Run the following command:

```
bin\etxsvr runtime add filename
```

To remove a runtime

Run the following command:

```
bin\etxsvr runtime remove version
```

To set the default runtime version

Run the following command:

```
bin\etxsvr runtime setdefault version
```

Applying a service pack update to Exceed TurboX Launcher

Exceed TurboX Client Launcher files are packaged inside each Exceed TurboX Server patch.

You can manually update the launcher on your system by downloading the latest version from the **User settings** dialog box on the Exceed TurboX web UI. For more information, see *Downloading and installing the Client Launcher* in *Exceed TurboX Dashboard and Client Help* or in *Exceed TurboX Server Manager Help*.

Notes:

In Exceed TurboX:

- You can use the launcher automatic upgrade capability for Windows and Linux platforms, if you are using an authentication method other than Kerberos.
- If you are currently using an older launcher version (which does not support Kerberos), and you want to enable Kerberos parameter support, you must uninstall the 12.0 launcher and manually install the 12.5.x launcher.
- The launcher automatic upgrade capability is not available for Mac platforms.

Installing the service pack on Exceed TurboX Server

This section provides a summary of how to install the service pack on Exceed TurboX Server on supported Linux and Solaris platforms.

To install the service pack on Exceed TurboX Server:

1. Apply the service pack update to Exceed TurboX Server (see Applying a service pack update to Exceed TurboX Server).

   The Server and the Connection Nodes are now running the latest version.
2. Install the runtime (see Installing a runtime using automated scripts).
3. Configure the Exceed TurboX Launcher.

Notes:

- The runtime version included in this Server patch is also added to the Runtime library available for your site.
- The latest version of the Client Launcher is available for download from the Exceed TurboX web UI.

Setting the default runtime

Exceed TurboX Client and Proxy updates are issued as runtime (`.ZIP`) files, ensuring that the Client and Proxy are always at the same patch level. Exceed TurboX Server can support multiple runtime versions, one of which must be designated as the default.

Applying a service pack update to Exceed TurboX Server does not automatically add the new runtime to the list of available runtimes on the Server, but you must manually set the new runtime as the default runtime, if you want it to be used.

To verify if the Exceed TurboX runtime included in the service pack is installed, sign in to the Exceed TurboX Server Manager as an administrator, and review the list of runtimes installed on your site (for more information, see *Managing runtimes* in the *Exceed TurboX Server Manager Help*.

To designate the newly-added runtime as the default version, see *Designating, disabling, and removing runtimes* in the *Exceed TurboX Server Manager Help*.

Applying other service pack updates

On Linux and UNIX platforms, the Exceed TurboX Connection Node package includes files that are not updated by the Server automatic patching process. These files are VirtualGL for Linux, keyboard files, font files, and the Space Ball utility. Customers who want these updated files must extract the Connection Node patches over the patched nodes.

This section describes how to apply service pack updates to keyboards, fonts, and third-party components on your Exceed TurboX site, on Linux and UNIX platforms.

> **Note**
>
> If you have installed Exceed TurboX Server on a Windows platform, these updates are provided in an `.msi` file and applied in the same way as an `.msi` installation. See Installing Exceed TurboX Server on a Windows platform for instructions.

To apply updates to keyboards, fonts, or VirtualGL components (on Linux/UNIX platforms):

1. Copy the `ETXCN-ETX Version.Build Number-SP Number-Platform-CPU.tar` file for your platform into the Exceed TurboX node directory.

   For example:

   `ETXCN-12.0.1.5790-SP1-linux-x64.tar`

2. Extract the files into this directory, then run the patch script.

   `./bin/patch`

   Any updates related to keyboards, fonts, or VirtualGL components are now applied to your Exceed TurboX connection nodes.

6. Understanding Kerberos concepts

Understanding Kerberos concepts

This section provides background information on single sign-on authentication (SSO) for Exceed TurboX, based on Kerberos. For more information on how to configuring your environment for SSO authentication, see Configuring your environment for single sign-on authentication based on Kerberos.

About Kerberos

Kerberos is a network authentication service developed for open network computing environments where simple password authentication cannot provide the required level of security. With Kerberos, passwords are never sent over the network. Instead, they are used to open a cryptographic key that is then used to encode and decode messages. If the password is incorrect, the key cannot be opened and messages remain indecipherable.

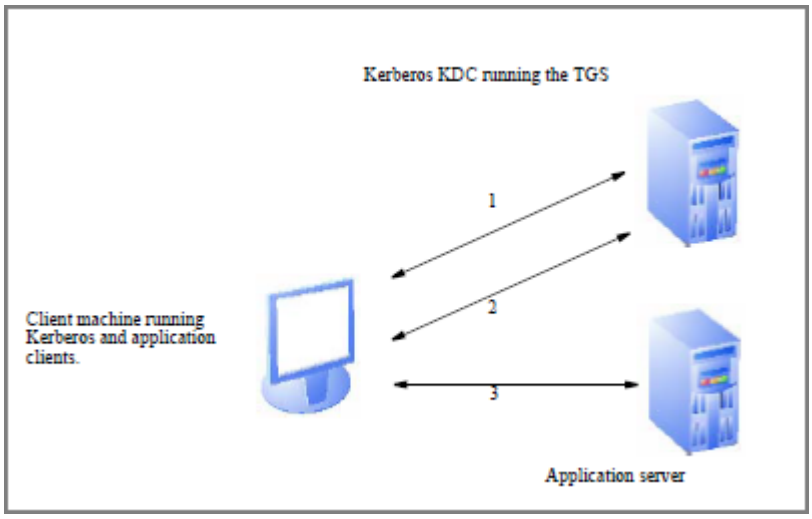
The Kerberos protocol is based on the Needham and Schroeder authentication protocol with the addition of timestamps to simplify basic server/client authentication, a Ticket Granting Service to support single sign-on, and its own method of cross-realm authentication.

Kerberos terminology and architecture

Kerberos terminology and architecture

The following diagram illustrates the Kerberos architecture:

1. When you start a Kerberos session, Kerberos connects to the Key Distribution Center (KDC) to acquire a session key and a Ticket Granting Ticket (TGT).
2. When you start an application, Kerberos connects to the Ticket Granting Service (TGS) to request a session key.
3. Now authenticated, the client sends its ticket and the session key to the application server to start a connection.



About Kerberos Principals (Kerberos Names)

A Kerberos principal is a unique name that identifies the party (typically a user or service) whose identity must be verified. A Kerberos name contains three parts:

- Principal name—This is usually a user or service name.
- Instance—In the case of a user, the instance is usually null. Some users may have privileged instances, however, such as "root" or "admin." In the case of a service, the instance is the name of the machine on which it runs. That is, there can be an rlogin service running on the machine ABC, which is different from the rlogin service running on the machine XYZ.
- Realm—This identifies the Kerberos service providing authentication for the principal.

When writing a Kerberos name, the principal name is separated from the instance (if not null) by a forward slash (/). The realm (if not the local realm) follows, preceded by the `@` sign. The following are examples of valid Kerberos names:

- `user1`
- `brobin.admin@LCS.com` (for Kerberos version 4)
- `brobin/admin@MAGIC.HCL.com` (for Kerberos version 5)
- `user1@MAGIC.HCL.COM`

About verifiers

Verifier refers to the party, usually a server or an application server, that requires confirmation of a principal's identity.

About Kerberos clients

The Kerberos client is any party on a network (typically a user or application) that requires a Kerberos ticket in order to be authenticated and permitted to access resources.

About Kerberos realms

A Kerberos realm consists of an Authentication Server and the users or servers it serves. An organization may have multiple Authentication Servers and, therefore, multiple Kerberos realms.

Key Distribution Centers

Key Distribution Centers

Every Kerberos realm must have one or more Key Distribution Centers (KDC). Each KDC is comprised of the following components:

- a database of principals, their keys, and other information
- an Authentication Server (AS)
- a Ticket Granting Service (TGS)

About Authentication Servers

The Authentication Server (AS) issues encrypted Ticket Granting Tickets (TGT) to clients that successfully log in to a realm.

About Ticket Granting Services

Ticket Granting Services (TGS) issue unique, service-specific tickets after receiving two types of information from the client making the request. The first type of data is a ticket request, which specifies the principal name of the service the client wants to access. The second is the Ticket Granting Ticket provided by the Authentication Server at login time. Ticket Granting Services issue service tickets only after confirming that the Ticket Granting Ticket presented by a client is encrypted with the Authentication Server's key.

About Ticket Granting Tickets

Ticket Granting Tickets (TGT) are issued by the Authentication Server upon successful login and allow the client to get additional tickets without providing login credentials again. They contain information including the name of the principal, a random session key and the session key's expiration time. They also contain information collectively referred to as the authenticator. When access to a resource is required by a client, the Ticket Granting Ticket is presented to the Ticket Granting Service, which then issues a service-specific ticket (server ticket) used to authenticate the client.

About the authenticator

The authenticator comprises part of a client's application request, the other part being the Kerberos ticket issued by the authentication server and used for authentication. The authenticator is a collection of information that is encrypted with the session key belonging to the related Kerberos ticket. The information includes the current time, a checksum, and an encryption key (optional).

When a verifier receives an application request, it decrypts the Kerberos ticket, thereby acquiring the session key, which it then uses to decrypt the authenticator. The verifier uses a checksum to ensure that the key used to encrypt the authenticator is the same key it used to decrypt it. If so, then the verifier has proof that the principal specified in the accompanying ticket generated the authenticator and that the principal is the owner of the session key.

The verifier also checks the timestamp to ensure that the authenticator was not intercepted and used later by an imposter. If the timestamp is within the allotted time and does not match the timestamp present in other requests, the verifier accepts the request.

About server tickets

About server tickets

Server tickets are issued by the Key Distribution Center (KDC). Server tickets verify the identities of the involved parties. Each Server ticket possesses a session key that both parties share to secure their communications. Ticket flags specify configuration settings and permissions such as ticket forwardability and expiration.

About ticket expiration

Privileged tickets, such as root instance tickets, expire in a few minutes, while tickets that carry more ordinary privileges may be good for several hours or a day, depending on the installation's policy. If your login session extends beyond the time limit, you will have to reauthenticate yourself to Kerberos to get new tickets.

About ticket sharing

Kerberos tickets are stored in the memory cache. Tickets are obtained when the user logs in to the environment. Thus, during normal operation, users enter a password once only. On some other platforms, however, the tickets are stored separately in each Kerberized application, and each time one of these Kerberized applications starts, the user must reenter the password.

About session keys

A session key is a unique encryption key with a limited life span generated by the Key Distribution Center (KDC) to authenticate a client. The session key is distributed to both the client and the verifier.

About Kerberized servers

A Kerberized server is an application server that supports Kerberized access with different Internet protocols, such as SSH or FTP. With a Kerberized client, you can connect to such a server securely, and you can also encrypt your session.

About cross-realm authentication

About cross-realm authentication

Users may require access to resources located in realms other than their local realms. Kerberos cross-realm authentication makes such access possible, though it does require some configuration.

About direct cross-realm configuration

In a direct cross-realm authentication scenario, remote Ticket Granting Service principals are created for each realm. To configure cross-realm authentication between realms AAA.COM and BBB.COM, for example, the principals `krbtgt/AAA.COM@BBB.COM` and `krbtgt/BBB.COM@AAA.COM` must be created on each of the two realms. Every pair of realms configured for cross-realm authentication must share a unique key, that of the Ticket Granting Service principal.

About certification path configuration

Whereas direct cross-realm authentication results in a full matrix of interconnected pairs of realms, certification path configuration is a hubandspoke or hierarchical model, and, therefore, more easily managed as it requires less key sharing.

> **Note**
>
> The Kerberos 4 protocol does not support the certification path model.

Certification path models use trusted intermediary realms to broker tickets that are implicitly trusted when presented to other realms for authentication. Every client and server must be appropriately configured to participate. Clients must use the certification path to contact the appropriate trusted Key Distribution Center (KDC). Servers use the certification path to ensure that the ticket presented for authentication was issued by a trusted KDC.

About KDC and service principals

Windows 2003 KDC maps a service principal name to a domain user account by using the `ktpass.exe` utility.

- The service principal names are stored in the `servicePrincipalName` attribute of the mapped domain user account.
- The service principal name format is standardized on all platforms and is of the form `serviceName/fullyQualifiedHostName@realmName.`
- Two interested `serviceName` types are `exceed` and `host`.
- Each instance of a service principal must be mapped to a unique user domain account using the `ktpass.exe` utility.

  The last keytab file is created with the increased KVNO and prior keytab file(s) contain an invalid KVNO resulting in the error `"Key version number for principal in keytable is incorrect."` The only correct/working KVNO is 3.

7. Glossary

application host

The remote computer that hosts the applications with which you want to work. An application host can be any supported platform type, such as Windows or Linux.

application

The core purpose of Exceed TurboX is to allow users to launch applications and desktops on a remote computer. When you launch a profile from Exceed TurboX Dashboard, a session starts up. Depending on the profile configuration, one or more applications or desktop sessions may launch. These applications are running on a remote host but are designed to appear as if they are running on the local client machine. X Window applications can also be referred to as X clients.

Client Menu

The Client Menu is available after you use Exceed TurboX Dashboard to launch a session. The Client Menu allows you to perform tasks within the sessions you start. The commands offered on the menu differ depending on the operating system of the client workstation. In most cases, you can use the menu to perform such tasks as:

- sharing and suspending the session
- terminating the session
- generating a trace

Connection Node

The processing hub of Exceed TurboX. The connection node acts as an intermediary between the client and the remote host and is responsible for managing the session, compressing the remote display, and handling input and other requests from the Exceed TurboX Client once a session has been established. Also called a proxy host.

Dashboard

This web-based interface is your access point to Exceed TurboX and your gateway to the remote applications you need to work with. You can use it to configure, launch, and manage sessions.

ETX RDP protocol

The protocol or language used to transfer information between the proxy and the Windows application or desktop host to which you connect. Windows applications communicate with the proxy using the ETX RDP protocol. In turn, the proxy communicates with your workstation using the Thin X Protocol (TXP).

etxlog.txt

Log file generated by Exceed TurboX for each session, which records detailed information about the session for troubleshooting purposes.

etxscan utility

The Exceed TurboX application scanner (etxscan) scans the connection node and returns a list of installed applications to Exceed TurboX Server. These applications can, in turn, be published. The etxscan utility provides information about installed applications from the XDG standard menu system and Windows Start menu, including the application path, parameters, and working directory. This allows Exceed TurboX Server to launch those applications remotely. On Windows, etxscan also supports a --syscheck argument to print GPU hardware support.

Exceed TurboX Client

The Exceed TurboX Client consists of two partsthe Client Launcher and the Client Runtime. The launcher downloads and executes the correct client runtime. The runtime provides all client-side functionality, such as launching and managing sessions, displaying the client-side menus, transferring files, and communicating with the remote host.

Multiple Window mode

Application and desktop profiles can be configured to run in either Multiple Window mode or Single Window mode. In Multiple Window mode, each application runs inside its own application window, as if the application was running natively on the users machine.

profile

Exceed TurboX profiles contain settings that define the look and behavior of Exceed TurboX sessions. You launch sessions from profiles listed on Exceed TurboX Dashboard. Administrators can create Group profiles to accommodate different session requirements. Users cannot edit Group profiles, but they can customize or copy and edit them for their own use. proxy host See Connection Node.

proxy

Each time a connection node receives a request to start a session, it starts a new proxy with a unique display ID. The proxy is responsible for communicating between your workstation and the remote application or desktop host. When you terminate your session, the server closes the associated proxy.

published application

Applications installed on a Windows or X Window host can be published in Exceed TurboX by means of an application scanner (etxscan) that is installed on the Exceed TurboX Connection Node. Exceed TurboX administrators can publish scanned applications to make them available to users.

resizing policy

Defines how the root window is resized.

*Fixed*A user or the system defines the size of the root window. If you click the Maximize button in the bar, the main window is restored to its original position and size. By default, the session uses Fixed resizing policy and is displayed on a primary monitor.

*Scaled*When you resize the window, the size of the root window does not change, but the contents of the session scale up/down accordingly. For example, if multiple sessions are running, you may view them as thumbnails. Scaling does not affect the behavior of UNIX applications, because the root window size remains the same.

*Dynamic*Allows you to resize the Exceed TurboX window dynamically. The application (such as the KDE environment) will reflect this and fit in the new window. For example, you may switch from Single to Multiple Window Mode or vice versa. The applications will renegotiate the new root window size and be redrawn appropriately.

REST

Representational State Transfer (REST) is a software architectural style that defines a set of constraints to use to create web services. Web services that conform to the REST architectural style are called RESTful web services; they provide interoperability between computer systems on the Internet. RESTful web services allow the requesting systems to access and manipulate textual representations of web resources by using a uniform and predefined set of stateless operations. In a RESTful web service, the requests made to a resources URI generate a response with a payload formatted in HTML, XML, JSON, or some other format. The response can confirm that some chanage has been made to the stored resource, and can provide hypertext links to other related resources or collections of resources. When HTTP is used, the following operations are available: GET, POST, PUT, DELETE, and other predefined CRUD HTTP methods.

REXEC protocol

The REXEC (Remote EXECute) protocol launches applications on a remote host. It requires a user ID, password, host address, and command to execute on the remote host. You can select this startup method in Xstart.

RLOGIN protocol

The RLOGIN protocol establishes a remote connection to run X applications. It allows an authorized user to sign in to hosts on a network and interact as if the user were physically at the host computer. You can select this startup method in Xstart.

root window

The parent (container) window which opens when launching a Single Window Mode profile. This root window contains either a remote desktop or multiple

remote application windows. It keeps all session windows in a single container, so that users can manage all windows as a single group and manage multiple sessions more easily. By contrast, In Multiple Window mode, application windows are opened directly on the users desktop as native windows, not within a root window.

RSH protocol

RSH is a protocol for executing commands on a remote host, passing it input and receiving its output. RSH communicates with a daemon on the remote host. A benefit of RSH is its ability to reference a file called .rhosts, which resides on the host and maintains a list of terminals allowed to sign-in without a password. You can select this startup method in Xstart.

Secure Shell protocol

Secure Shell (SSH) is a TCP-based protocol that provides authentication, encryption, and data integrity security features. In Exceed TurboX, SSH provides a secure channel between the connection node and application host, for sending and receiving user inputs and display protocol. You can select this startup method in Xstart.

Server Manager

This is the web-based administration interface for Exceed TurboX. session A connection to another computer, established by Exceed TurboX, that moves information (including keyboard input and screens, for example) between them. You configure and launch these connections with Exceed TurboX to work with one or more applications installed on the computer that you connect to. You can use Exceed TurboX Dashboard to start, manage, and end sessions.

Single Window mode

Application and desktop profiles can be configured to run in either Multiple Window mode or Single Window mode. In Single Window mode, each application runs inside the root window, which is a single window that contains all of the remote applications. Applications running inside the root window may overlap each other and cannot be moved outside of the root window.

taskbar icon

This icon appears on your desktop taskbar when you start or join a session. You may have multiple icons in the taskbar, each representing a session that you have started. Joined sessions are always displayed in Single Window mode and represented by one icon only. Right-click the sessions taskbar icon to access the Client Menu.

template

Administrators create and configure templates for users who need to create profiles of their own. Users create profiles by copying existing templates. The administrator-specified settings in the template are copied to the profile; individual settings in the template can be marked as read-only so that they cannot be changed by users. This provides the administrator with complete control over which settings users can modify.

When an administrator modifies a template, the change is reflected in all profiles that are based on that template.

Windows

The Microsoft Windows operating system should not be confused with the X Window System. Exceed TurboX includes the capability to run remote sessions by connecting to either an X Window (UNIX or Linux) or Microsoft Windows remote desktop or application host.

X application

Any application that uses the X Window System to draw its graphical user interface. Although X applications are written primarily on the UNIX and Linux operating systems, it is possible to create X applications that run on other platforms, such as Microsoft Windows.

X display

Each time you start an X Window session, an X server is started on the host to which you connect. The X display identifies this specific X server which will be used to manage your session (for example, by transferring input from your mouse and keyboard to the application). The applications you work with are aware of which X display you are using. An X display is referenced using the following notation: `<HostIP>:<Display#>`, where `<Display#>` is incremented for each new session started on that host.

X protocol

One of the protocols used to transfer information between your computer and the X Window application host that you connect to. In Exceed TurboX, applications communicate with the proxy using the X protocol. In turn, the proxy communicates with your workstation using the Thin X Protocol (TXP).

X selection

The text or other data, such as an outlined region of the screen, that you have selected for copying and pasting to another open window in either the same session, another session, or on your workstation.

X server

An intermediary component that Exceed TurboX launches to handle communication such as key and data transfer (visual screens and windowing) between your workstation and the application host. The X server is composed of both the software launched to handle your session and the hardware (mouse, keyboard, monitors) used to communicate and display screens. The X server also handles font rendering and resource management.

X Window Manager

An X Window Manager is a program that provides basic management commands for application windows, including opening, closing, moving, and resizing windows. Most window managers are installed with the operating system. The X Window Manager handles all window functions and often provides a menu from which you can select commands to start other applications. The window manager you use can be installed on your workstation or on a remote machine. You must start the window manager. It does not start by default. You can set window manager options on the Display tab (Basic mode) and Window mode tab (Advanced mode) when you create or edit profiles.

X Window

When running applications on a UNIX or Linux host, application windows are rendered using X drawing primitives such as lines and rectangles. These application windows are referred to as X Windows. Depending on the operating system, a different X Window Manager may be used, which affects the appearance and functionality of your application windows. An X Window can be rendered on the host or on the client desktop.

Xstart

Custom startup profiles typically contain one or more applications or commands. These applications or commands are called Xstarts. Xstarts allow you to specify:

- the application host to connect to
- your sign-in credentials for that host
- command line parameters for the application
- additional options such as user prompts and advanced flags

You can configure multiple Xstarts within a profile.

8. Notices

Copyright

© 1996-2025 Rocket Software, Inc. or its affiliates. All Rights Reserved.

Trademarks

Rocket is a registered trademark of Rocket Software, Inc. For a list of Rocket registered trademarks go to: www.rocketsoftware.com/about/legal. All other products or services mentioned in this document may be covered by the trademarks, service marks, or product names of their respective owners.

Examples

This information might contain examples of data and reports. The examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

License agreement

This software and the associated documentation are proprietary and confidentical to Rocket Software, Inc. or its affiliates, are furnished under license, and may be used and copied only in accordance with the terms of such license.

Note: This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when exporting this product.

Corporate information

Rocket Software, Inc. develops enterprise infrastructure products in four key areas: storage, networks, and compliance; database servers and tools; business information and analytics; and application development, integration, and modernization.

Website: www.rocketsoftware.com

Contacting Technical Support

The Rocket Community is the primary method of obtaining support. If you have current support and maintenance agreements with Rocket Software, you can access the Rocket Community and report a problem, download an update, or read answers to FAQs. To log in to the Rocket Community or to request a Rocket Community account, go to www.rocketsoftware.com/support. In addition to using the Rocket Community to obtain support, you can use one of the telephone numbers that are listed above or send an email to support@rocketsoftware.com.

Rocket Global Headquarters
77 4th Avenue, Suite 100
Waltham, MA 02451-1468
USA

Country and Toll-free telephone number

To contact Rocket Software by telephone for any reason, including obtaining pre-sales information and technical support, use one of the following telephone numbers.

- United States: 1-855-577-4323
- Australia: 1-800-823-405
- Belgium: 0800-266-65
- Canada: 1-855-577-4323
- China: 400-120-9242
- France: 08-05-08-05-62
- Germany: 0800-180-0882
- Italy: 800-878-295
- Japan: 0800-170-5464
- Netherlands: 0-800-022-2961
- New Zealand: 0800-003210
- South Africa: 0-800-980-818
- United Kingdom: 0800-520-0439