Exceed TurboX

Authentication Server

V12.5.4

Table of Contents

1. Introduction

Rocket® Exceed TurboX (ETX) is a web-based platform that enables users to launch UNIX, Linux, and Windows desktops and applications over the internet. ETX 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 provides conceptual and task-oriented information about how to install and manage the ETX Authentication Server. This document is intended for ETX administrators.

2. Installing the Authentication Server

Installing the ETX Authentication Server

The ETX Authentication Server is the component that provides different types of authentication options for the ETX Server, specializing in those that involve Web Authentication such as OpenID Connect (OIDC), SAML (Security Assertion Markup Language) to authentication providers such as Okta or Microsoft.

The Authentication Server is only supported on Linux base platforms and communicates with different identity providers to provided ETX Sites with authentication.

Installing the Authentication Server on Linux

This section describes how to install the Authentication Server on Linux platforms.

Requirements

Linux running on x86_64 or aarch64 architectures.

Installing the Authentication Server

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 `etxauthsvr` for the installation files:

   ```
   $ sudo mkdir /opt/etxauthsvr
   ```

   Type your password at the prompt.
4. Use the `sudo` command to create an `etxauthsvr` 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/etxauthsvr
   ```
6. Extract the files from the installation package into the `/opt/etxauthsvr` directory:

```
$ sudo tar xzvf <path>/<packagename>.tar.gz
```

where:

- `<path>` is the path to the `ETXAuthServer` directory on the product installation media.
- `<packagename>` is the installation file corresponding with your system (`ETXAuthServer-Version.Build Number-ServicePackNumber-HotfixNumber-Platform-CPU.tar.gz`).

The following example is the Linux package name for version 12.5.4, build 9887 representing Hotfix 3 for Service Pack 1:

```
ETXAuthServer-12.5.4.9887-SP1-HF3-linux-x64.tar.gz
```

7. Enter your password, if you are prompted.

   The Authentication Server is now installed.

8. Before starting the Authentication Server, it must be configured. Do one of the following:

   - Start the server using the default pre-configured settings provided with the product. See Starting and stopping ETX Authentication Server on a Linux platform.
   - Customize the Server settings before starting the Server. See Configuring ETX Authentication Server settings.

Patching the Authentication Server on Linux

This section provides the standard installation procedure for the Authentication Server. Some hotfixes may use a slightly different installation procedure (listed in their description). The versions listed below are examples only.

About the installation package

Authentication Server installation patch package names use the following naming convention:

```
<component prefix>-<ETX version>.<build#>-[SP<#>-]HF<#>-update-<platform>.
```

The prefix stands for the Authentication Server (`etxauthsvr` prefix).

Example:

```
etxauthsvr-12.5.4.9yyy-HFz-update-linux-x64.sh
```

> **Note**
>
> File names used as examples in these installation procedures are from Hotfix z. To view the list of files updated for subsequent hotfixes, see the *Files delivered* section for each hotfix.

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

Updating the Exceed TurboX Authentication Server

1. Copy the `.sh` file appropriate to your platform to the ETX Server directory.
2. Ensure the `.sh` file has execute permissions set.
3. During patching, the ETX Server must be stopped:

```
$ cd /opt/etx/etxauthsvr
$ bin/etxauthsvr stop
```

4. Run the shell script from the ETX Server directory as the user who originally installed the Server. That is, as the user that owns the `data/etxdata` folder. For example:

```
$ cd /opt/etx/etxauthsvr
$ sh ./etxauthsvr-12.5.4.9yyy-HFz-update-linux-x64.sh
```

5. The script determines if the contents of the patch need to be applied to this ETX Server directory. If there is applicable content then the script prompts you:

```
Do you want to continue

[y]es or [n]o (<Enter> for no)

Type `y` to continue.
```

   The script runs, applying the update.

6. After the script completes you can start the server again:

```
$ bin/etxauthsvr start
```

3. Managing the Authentication Server

Managing the Authentication Server

The Authentication Server is the component that provides different types of authentication options for the ETX Server, specializing in those that involve Web Authentication such as OpenID Connect (OIDC), SAML (Security Assertion Markup Language) to authentication providers such as Okta or Microsoft.

The Authentication Server is only supported on Linux base platforms and communicates with different identity providers to provied ETX Sites with authentication.

Basic configuration required for the Authentication Server

The Authentication Server requires a few settings to be changed prior to use. This section describes these basic configuration choices required before starting the server.

The `config` and `authtype` commands are the primary commands that must be run before starting the server. The `export` command is useful for providing values to your Server Authentication settings.

AuthType Selection

The Server needs a `data/conf/authtype` to use when starting. To establish this content, the `authtype` command must be run. See Changing the AuthType from the command line.

Port Selection (optional)

The Server will default to use port 9443. You can use `config httpsPort=443` to change the port. See Using etxauthsvr config from the command line.

Application.properties changes

After the AuthType has been configured, the main changes required relate specifically to the Identity Provider you will be using. These are contained inside the `data/conf/authtype/application.properties` file. It is recommended to change only the values required. See Managing AuthTypes.

Keystore preparation

All communication coming into the Authentication Server and going out from it to other Identity Providers is based on secured TLS connections. A proper keystore for both incoming and outgoing connections must be established. See Configuring keystores for the Authentication Server.

Run ETX Authentication Server as a service (optional)

You can register and control the Authentication Server as a service. This is not required for the normal functioning of the Authentication Server. See Configuring the Authentication Server as a Service.

Configure your ETX Site to use ETX Authentication Server

When your Authentication Server is ready for use. You can export the settings required for your ETX Site to use. See Exporting the Authentication Server settings.

Using etxauthsvr config from the command line

You can configure the Authentication Server settings by running the `bin/etxauthsvr` script.

Several key settings to be shared between different Authentication Type application properties are controlled by the `config` command.

You can use `bin/etxauthsvr config key=value` to adjust most of these settings.

This section provides a list of settings you can use to configure the Authentication Server on Linux platforms.

Key	Value
`httpsPort`	Specifies the port number to be used by ETX Auth Server for TLS-encrypted HTTPS connections. Default value is `9443`.
`authsvc.logging.level`	Specifies the level of detail to be included in ETX Authentication Server logs. Possible values are: `TRACE`, `DEBUG`, `INFO` (default value), `WARN`, and `ERROR`.
`maxJavaHeapSize`	Specifies the maximum size of the Java heap as a percentage of total system memory. Default value is `50%`.
`logFilter`	Optional. Specifies the Java filter string for the main Authentication Server log.
These are trust store related:	
`authsvc.tlstrust.key-store`	File name to contain public certificate chain for outgoing client TLS connections to Identity provider (default: `truststore.bcfks`).
`server.ssl.trust-store`	Future alternative client REST API.
`authsvc.signing.key-store-password`	Password to read the `authsvc.tlstrust.key-store` file (default: `changeit`).

Key	Value
`server.ssl.key-store`	File name to contain public and provide key information for the Authentication Server to provide TLS connections to clients (default: `keystore.bcfks`).
`server.ssl.key-store-password`	Password to read the `server.ssl.key-store` file (default: `changeit`).
These are Auth Type related:	
`authtype`	Lists the current authentication type that the Authentication Server is supporting (read-only).
`etxserver.client-id`	Client ID string for the ETX Server to connect to this Authentication Server (default: `hc-client`).
`etxserver.client-secret`	Client secret string for the ETX Server to connect to this Authentication Server (default: `hc-client-secret`).
`etxserver.trusted.baseurllist`	Lists the list of ETX Server base URLs that are permitted to be redirected back to from this Authentication Server.

Changing a config value

You can change a config value by using the `config key=value` command.

Example showing how to change the listening port for the Authentication Server:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr config httpsPort=443
Server setting 'httpsPort' has updated to '443' successfully.
$
```

Listing config values

You can print the current config value by using the `config key` command.

Example of listing a single config value:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr config httpsPort=443
Server setting 'httpsPort' has updated to '443' successfully.
$
```

> **Note**
>
> By omitting the key value from the `config` command, you are shown the complete list of current keys and values.

Controlling ETX Site Redirection

The `config` command also supports `-addserver` and `-delserver` sub-commands. These are specifically used to manage the `etxserver.trusted.baseurllist` config value. This value is important to ensure that the Authentication Server is able to redirect the client browser back to the correct ETX Sites. Several ETX Servers or ETX Sites can share the same Authentication Server, so you might be required to add or remove ETX base URL paths.

Adding an ETX Site

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr config -addserver https://myetxserver.fqdn:8443
Added URL: https://myetxserver.fqdn:8443
Successfully updated server.properties: /opt/etxauthsvr/data/conf/
server.properties
Verification successful: https://myetxserver.fqdn:8443
$
```

Removing an ETX Site

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr config -delserver https://myetxserver.fqdn:8443
Removed URL: https://myetxserver.fqdn:8443
Successfully updated server.properties: /opt/etxauthsvr/data/conf/
server.properties
Verification successful:
$
```

Configuring keystores for the Authentication Server

There are two main keystores used by the Authentication Server.

The first keystore is for incoming TLS connections and is represented by the `server.ssl.key-store` and corresponding `server.ssl.key-store-password` config values.

The second keystore is for outgoing client TLS connections to the selected Identity Provider and is represented by the `authsvc.tlstrust.key-store` and corresponding `authsvc.signing.key-store-password` config values.

You can see how to change these keystores by reading Using etxauthsvr config from command line.

Establishing the keystore

When the Authentication Server is first extracted, there is no incoming `keystore.bcfks`. If you start the server for the first time, one is automatically created for you.

You can also use the `keystore` command to create the file as follows:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr keystore create
Enter keystore password:
Generating Keystore file...
Keystore generated successfully: /opt//etxauthsvr/data/ssl/keystores/
keystore.bcfks
Successfully generated keystore
```

Cloning the ETX Server cerificate

If you are running the Authentication Server on the same server as your ETX Server you can use the following steps to clone the keys into the `keystore.bcfks`.

> **Note**
>
> In the following example, the password used for the keystore is `changeit`. It should be adjusted to the correct password you will be using.

1. Convert the `.PEM` files from the ETX Server into single P12 file:

```
cd /opt/etxsvr/etxsvr-12.0/data/ssl
openssl pkcs12 -export -in server.crt -inkey server.key -out good-
keystore.p12 -name servlet-engine -passout pass:changeit
```

2. Prepare the `data/ssl/keystores` folder and import the P12 as a `BCFKS` file:

```
cd /opt/etxauthsvr/data/ssl/keystores
rm keystore.bcfks etxauthsvr.crt
/opt/etxauthsvr/lib/jdk/bin/keytool -importkeystore -srckeystore /opt/etxsvr/
etxsvr-12.0/data/ssl/good-keystore.p12 -srcstoretype JKS -srcstorepass
changeit -srcalias servlet-engine -destkeystore keystore.bcfks -deststoretype
BCFKS -destalias servlet-engine -deststorepass changeit -provider
org.bouncycastle.jce.provider.BouncyCastleProvider -providerpath/opt/
etxauthsvr/lib/java/bcprov-jdk15on-1.68.jar
```

3. Change the password the Authentication Server will use to read the file:

`cd /opt/etxauthsvr` `bin/etxauthsvr config authsvc.signing.key-store-password=changeit`

Establishing the truststore.bcfks file

When the Authentication Server is first extracted, there is a default `truststore.bcfks` file. It contains several example Microsoft Azure certificates. The intention is to replace these with the certificate chain from the Identity provider you will use.

There are many formats that you might have for your certificate chain. The Authentication Server provides the JDK keytool which can be used to perform many operations.

Example of importing a P12 keystore into the BCFKS `truststore.bcfks` file:

```
cd /opt/etxauthsvr/data/ssl/keystores
/opt/etxauthsvr/lib/jdk/bin/keytool -v -importkeystore -srckeystore /
somelocation/keystore.p12 -srcstoretype PKCS12 -srcstorepass changeit  -
destkeystore truststore.bcfks -deststoretype BCFKS -deststorepass changeit -
provider org.bouncycastle.jce.provider.BouncyCastleProvider -providerpath /
opt/etxauthsvr/lib/java/bcprov-jdk15on-1.68.jar
```

Notes

- You might have to provide a `-srcalias source_alias` argument as well if the source keystore has an alias defined. The destination alias should remain blank.
- You will have to change the password from `changeit` to whatever password is required.

Configuring the Authentication Server as a Service

This section describes how you can register the Authentication Server as a service and allow it to start up with the system boot.

Making the Service

You can register the Authentication Server as a service value by using the `bootstart enable` command. Once a service has been registered, it will start when the system starts and can be controlled by the appropriate linux service command.

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

If there is more than one instance of extsvr on the computer, the name of the service has the format `etxauthsvr_n`, where `n` is a unique identifying number. For example, `etxauthsvr_2`.

Example of enabling the service:

```
$ cd /opt/etxauthsvr
$ sudo bin/etxauthsvr bootstart enable
[sudo] password for username:
Enabling ETX Authentication Server  etxauthsvr  boot service.
Created symlink /etc/systemd/system/multi-user.target.wants/
etxauthsvr.service → /etc/systemd/system/etxauthsvr.service.
Bootstart status: on
Service name    : etxauthsvr
User account    : username
```

Removing the Service

You can unregister the ETX Authentication Server as a service value by using the `bootstart disable` command. Once a service has been unregistered, it will no longer start when the system starts and can no longer be controlled by the appropriate linux service command.

Example of removing the service:

```
$ cd /opt/etxauthsvr
$ sudo bin/etxauthsvr bootstart disable
[sudo] password for username:
Removed /etc/systemd/system/multi-user.target.wants/etxauthsvr.service.
ETX Authentication Server bootstart is off.
```

Checking the service status

To see the current state of the service, you can use the `bootstart status` command.

Example of checking the service status when it is disabled:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr bootstart status
[sudo] password for username:
ETX Authentication Server bootstart is off.
```

Starting and stopping the Authentication 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 Authentication Server:

- Check the status of the Authentication Server.

  The Authentication Server is not accessible to the ETX Server if it is not running. You may need to check the status of the Authentication Server if you are unable to access the Dashboard or Server Manager login page.

- Start the Authentication Server.

  You can run the Authentication Server as an application or as a service. Starting the Authentication Server as a service allows the Authentication Server to start when the machine starts (the computer on which the Authentication Server is installed).

  Running the Server as an application ties the state of the Authentication Server with the user's session. However, running the Server as an application is useful for short-term tests and configuration. See Configuring the Authentication Server as a Service.

- Stop the Authentication Server.

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

Checking Authentication Server status

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

   `cd /opt/etx/etxauthsvr`
3. Do one of the following:

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

   `bin/etxauthsvr status`

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

   `systemctl status etxauthsvr.service`

   or

   `systemctl status etxauthsvr_n`

Starting the Authentication Server

Starting 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 Authentication Server installation folder.

   `$ cd /opt/etxauthsvr`
4. Enter the following command to determine the service name:

```
$ bin/etxauthsvr bootstart status -v
Bootstart status: on
Service name    : etxauthsvr
User account    : username
```

   The command prints the name of the `systemd` service it created.

   If there is more than one instance of `extsvr` on the computer, the name of the service has the format `etxauthsvr_n`, where `n` is a unique identifying number. For example, `etxauthsvr_2`.
5. Enter the following command to start the service:

   `systemctl start etxauthsvr`

or:

```
systemctl start etxauthsvr_n
```

Starting the Server as an application

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

   ```
   cd /opt/etxauthsvr
   ```
3. Enter the following command:

   ```
   bin/etxauthsvr start
   ```

Stopping the Server

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

   ```
   cd /opt/etxauthsvr
   ```
3. Do one of the following:

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

     ```
     bin/etxauthsvr stop
     ```
   - If the Server is running as a service, enter the following command:

     ```
     systemctl stop etxauthsvr
     ```

     or:

     ```
     systemctl stop etxauthsvr_n
     ```

> **Note**
>
> When the server is stopped, no ETX Site is able to redirect users to this Server, so Authentication is not possible.

Exporting the Authentication Server settings

Once your Authentication Server is ready to use with an ETX Site, you can use the `export` command to print the settings or save them to a JSON text file.

Exporting settings to an ETX Site

Generate settings to be used by the ETX Server to authentication with this Authentication Server:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr export
ETX Auth Server URL: https://yourserver.fqdn:443
ETX Auth Server Client id: hc-client
ETX Auth Server Client secret: hc-client-secret
Server Certificate PEM:
-----BEGIN CERTIFICATE-----
... public certificate data here...
-----END CERTIFICATE-----
```

If you want to send the settings to someone, you can export to a JSON text file file.

Exporting settings to a text file

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr export -f settings.json
Successfully saved to file: /opt/etxauthsvr/run/settings.json
```

Exporting settings to JSON

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr export -json
{
"ETX Auth Server URL": "https://yourserver.fqdn:443",
"Server Certificate PEM": "-----BEGIN CERTIFICATE-----\n... public
certificate data here...\n-----END CERTIFICATE-----\n",
"ETX Auth Server Client id": "hc-client",
"ETX Auth Server Client secret": "hc-client-secret"
}
```

Working with AuthTypes

Working with AuthTypes

The Authentication Server is the component that provides different types of authentication options for the ETX Server, specializing in those that involve Web Authentication such as OpenID Connect (OIDC), SAML (Security Assertion Markup Language) to authentication providers such as Okta or Microsoft.

The Authentication Server supports many Identity Providers and authentication methods. The following sections describe how to use the preset authtypes the Authentication Server supports:

- The Entraid AuthType
- The OKTA AuthType
- The LDAP AuthType
- The SAML AuthType

The Entraid AuthType

Registering the application

1. In a browser, log into the Azure Portal and navigate to `Home > Entra ID`.
2. Click `+ Add > App Registration`.
3. On the **Register an Application** page, enter the values for:
4. Name: `<NAME>`. For example, `ETX-Auth-yourapp`

Redirect URI Platform: `Web`

6. Redirect URI: `<ETX Server Auth URL>`. For example, `https://yourauthvm.eastus.cloudapp.azure.com:9443/osp/a/hc/auth/oauth2/landingpad`
7. Click **Register**.
8. In the left navigation tree, click `Manage > Authentication`.

Confirm that **Web Redirect URIs** contains the value that was just entered for your Authentication Server.

Creating the client secret

1. In the left navigation tree, click `Manage > Certificates & Secrets`.
2. Click **+ New Client Secret**.
3. On the **Add a client secret** panel, enter the **Description**, select **Expiration date duration** then click **Add**.

> **Notes**
>
> - Client secret values cannot be viewed, except for immediately after creation. Once created, be sure to save the secret before leaving the page.
> - The Secret Value for use during the Authentication Server configuration:
>
>   Sample:
>
>   - Value: `11111~22226666444888555555000002222`
>   - Secret ID: `12345678-1234-1234-1234-614856d5accb`

Adding an Application Owner

The users listed here can view and edit this application registration. Additionally, any user (might not be listed here) with administrative privileges to manage any application (for example, Global Administrator, Cloud App Administrator) can view and edit the application registrations.

1. On the left navigation tree, click **Manage > Owners**.
2. Click on **Add Owners**.
3. Search for or scroll to find the users, check the checkbox to the left of their name, then click **Select**.
4. Ensure that the expected user shows as the Owner.

Assigning Users to the application

1. Assign users and groups, apply conditional access policies, configure single sign-on, and more in Enterprise applications. In the left navigation tree, navigate to **Overview > Get Started > Configure for your organization > Go to Enterprise applications**.
2. Click on **Assign users and groups** (under **1. Assig-n users and groups**).
3. Click on **+ Add user/group**.
4. Click **None selected**.
5. Search for or scroll to find the users, check the checkbox to the left of their name, then click **Select**.

   Ensure that the number of users selected is the expected value.
6. Click **Assign**.

   Ensure that the users shown are the expected users.

Collecting values for the Authentication Server application.properties file

On the left navigation tree, navigate to **Overview**.

Alternatively, you can navigate to **Home > App Registrations >** which opens the **Overview** page for that App Registration.

Note the following two field values. They will be used when configuring ETX and the Authentication Server.

- **Application (client) ID**
- **Directory (tenant) ID**

You will also need the Client Secret Value that was created earlier, **Client Secret Value**.

Changing the Authentication Server to use Entraid

1. Enter:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr authtype enable entraid
```

2. Edit the now created `data/conf/entraid/application.properties` file:

```
$ vi data/conf/entraid/application.properties
```

Change the following values:

Configuration Field	Value
authsvc.oidc.client.id	Application (client) ID
authsvc.oidc.client.secret	Client Secret Value
authsvc.oidc.provider.url	Directory (tenant) ID

Note

For the authsvc.oidc.provider.url field, update only the tenant part that is after the login.microsoftonline.com section. Leave the rest unchanged. It should look like the following:

`https://login.microsoftonline.com/{tenant}/v2.0`

Starting and testing the Authentication Server

1. Enter:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr start
```

2. In a browser, navigate to: `https://<server>:9443/osp/a/hc/auth/app/`

   Example:

   `https://<authentication-server>:9443/osp/a/hc/auth/app/`

3. You may be prompted to accept App permissions or approve sign in request. Once accepted, you should get the `Session Authenticated` message.

Note

Once you receive a session authenticated message, this Authentication Server can be used by an ETX Site. You can configure the ETX Site to use this Authentication Server, then test the login.

Exporting your Authentication Server settings to your ETX Site

Generate settings to be used by ETX Server to authentication with this ETX Authentication Server:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr export
ETX Auth Server URL: https://yourserver.fqdn:443
ETX Auth Server Client id: hc-client
ETX Auth Server Client secret: hc-client-secret
Server Certificate PEM:
-----BEGIN CERTIFICATE-----
... public certificate data here...
-----END CERTIFICATE-----
```

1. Go to the **Site Settings -> Authentication page**.
2. In your ETX Site, log into the Server Manager.
3. Change the `Authentication Type` to `ETX Authentication Server`.
4. You can copy and paste the values from the `export` above into the fields on this page.
5. Press **TEST** to ensure communication to the Authentication Server is correct.
6. Press **SAVE**.

The OKTA AuthType

Register the application

1. In a browser, log into the Azure Portal and navigate to `Home > Entra ID`.
2. Click `+ Add > App Registration`.
3. On the **Register an Application** page, enter the values for:
4. Name: `<NAME>`. For example, `ETX-Auth-yourapp`

Redirect URI Platform: `Web`

6. Redirect URI: `<ETX Server Auth URL>`. For example, `https://yourauthvm.eastus.cloudapp.azure.com:9443/osp/a/hc/auth/oauth2/landingpad`
7. Click **Register**.
8. In the left navigation tree, click `Manage > Authentication`.

Confirm that **Web Redirect URIs** contains the value that was just entered for your Authentication Server.

Creating the client secret

1. In the left navigation tree, click `Manage > Certificates & Secrets`.
2. Click **+ New Client Secret**.
3. On the **Add a client secret** panel, enter the **Description**, select **Expiration date duration** then click **Add**.

> **Notes**
>
> - Client secret values cannot be viewed, except for immediately after creation. Once created, be sure to save the secret before leaving the page.
> - The Secret Value for use during the Authentication Server configuration:
>
>   Sample:
>
>   - Value: `11111~4444222266666668888888899999`
>   - Secret ID: `12345678-1234-1234-1234-bbbdddbbb222`

Adding an Application Owner

The users listed here can view and edit this application registration. Additionally, any user (might not be listed here) with administrative privileges to manage any application (for example, Global Administrator, Cloud App Administrator) can view and edit the application registrations.

1. On the left navigation tree, click **Manage > Owners**.
2. Click on **Add Owners**.
3. Search for or scroll to find the users, check the checkbox to the left of their name, then click **Select**.
4. Ensure that the expected user shows as the Owner.

Assigning Users to the application

1. Assign users and groups, apply conditional access policies, configure single sign-on, and more in Enterprise applications. In the left navigation tree, navigate to **Overview > Get Started > Configure for your organization > Go to Enterprise applications**.

2. Click on **Assign users and groups** (under **1. Assig-n users and groups**).
3. Click on **+ Add user/group**.
4. Click **None selected**.
5. Search for or scroll to find the users, check the checkbox to the left of their name, then click **Select**.

   Ensure that the number of users selected is the expected value.
6. Click **Assign**.

   Ensure that the users shown are the expected users.

Collecting values for the Authentication Server application.properties file

On the left navigation tree, navigate to **Overview**.

Alternatively, you can navigate to **Home > App Registrations >** which opens the **Overview** page for that App Registration.

Note the following two field values. They will be used when configuring ETX and the Authentication Server.

- **Application (client) ID**
- **Directory (tenant) ID**

You will also need the Client Secret Value that was created earlier, **Client Secret Value**.

Changing the Authentication Server to use OKTA

1. Enter:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr authtype enable okta
```

2. Edit the now created `data/conf/okta/application.properties` file:

```
$ vi data/conf/okta/application.properties
```

   Change the following values:

Configuration Field	Value
authsvc.oidc.client.id	Application (client) ID

Configuration Field	Value
authsvc.oidc.client.secret	Client Secret Value
authsvc.oidc.provider.url	Directory (tenant) ID

Note

For the authsvc.oidc.provider.url field, update only the tenant part that is after the login.microsoftonline.com section. Leave the rest unchanged. It should look like the following:

`https://login.microsoftonline.com/{tenant}/v2.0`

Starting and testing the Authentication Server

1. Enter:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr start
```

2. In a browser, navigate to: `https://<server>:9443/osp/a/hc/auth/app/`

   Example:

   `https://<authentication-server>:9443/osp/a/hc/auth/app/`

3. You may be prompted to accept App permissions or approve sign in request. Once accepted, you should get the `Session Authenticated` message.

Note

Once you receive a session authenticated message, this Authentication Server can be used by an ETX Site. You can configure the ETX Site to use this Authentication Server, then test the login.

Exporting your Authentication Server settings to your ETX Site

Generate settings to be used by ETX Server to authentication with this ETX Authentication Server:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr export
ETX Auth Server URL: https://yourserver.fqdn:443
ETX Auth Server Client id: hc-client
ETX Auth Server Client secret: hc-client-secret
Server Certificate PEM:
-----BEGIN CERTIFICATE-----
... public certificate data here...
-----END CERTIFICATE-----
```

1. Go to the **Site Settings -> Authentication page**.
2. In your ETX Site, log into the Server Manager.
3. Change the `Authentication Type` to `ETX Authentication Server`.
4. You can copy and paste the values from the `export` above into the fields on this page.
5. Press **TEST** to ensure communication to the Authentication Server is correct.
6. Press **SAVE**.

The LDAP AuthType

> **Note**
>
> ETX Server supports LDAP directly, but the Authentication Server can also be configured for LDAP use.

Changing the Authentication Server to use LDAP

1. Enter:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr authtype enable ldap
```

2. Edit the now created `data/conf/ldap/application.properties` file:

```
$ vi data/conf/ldap/application.properties
```

Example Microsoft AD settings

> **Note**
>
> The following values are examples only. You must update them to match your LDAP environment.

Change the following values:

Configuration Field	Value
authsvc.ldap.0.directory-type	AD
authsvc.ldap.0.search-user-base-dn	CN=users,DC=example,DC=org
authsvc.ldap.0.port	389
authsvc.ldap.0.tls-enabled	true
authsvc.ldap.0.directory-type-plugin-java-class	com.microfocus.emc.auth.config.LDAPStorePluginGeneric
authsvc.ldap.0.user-filter	person
authsvc.ldap.0.host	LDAP server host name or IP address
authsvc.ldap.0.attribute-login	uid
authsvc.ldap.0.admin-password	password
authsvc.ldap.0.admin-dn	admin
authsvc.ldap.0.attribute-guid	entryUUID
authsvc.ldap.enabled	true

Starting and testing the ETX Authentication Server

1. Enter:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr start
```

2. In a browser, navigate to: `https://<server>:9443/osp/a/hc/auth/app/`

   Example:

   `https://<authentication-server>:9443/osp/a/hc/auth/app/`

3. You will be prompted to enter your LDAP credentials. Once accepted, you should get the `Session Authenticated` message.

> **Note**
>
> After you receive a session authenticated message, this Authentication Server can be used by an ETX Site. You can configure the ETX Site to use the Authentication Server, then test the login.

Exporting your Authentication Server settings to your ETX Site

Generate settings to be used by ETX Server to authentication with this ETX Authentication Server:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr export
ETX Auth Server URL: https://yourserver.fqdn:443
ETX Auth Server Client id: hc-client
ETX Auth Server Client secret: hc-client-secret
Server Certificate PEM:
-----BEGIN CERTIFICATE-----
... public certificate data here...
-----END CERTIFICATE-----
```

1. Go to the **Site Settings -> Authentication page**.
2. In your ETX Site, log into the Server Manager.
3. Change the `Authentication Type` to `ETX Authentication Server`.
4. You can copy and paste the values from the `export` above into the fields on this page.
5. Press **TEST** to ensure communication to the Authentication Server is correct.
6. Press **SAVE**.

The SAML AuthType

This example guides you through the steps required to configure the Authentication Server to use SAML authentication.

Prerequisites

- Configure a SAML Identity Provider (IdP) that will be used to authenticate users.
- On the IdP platform, create an application that represents the Authentication Server.
- Collect the IdP metadata from your SAML provider. This is usually in XML format.

Collecting values for the Authentication Server application.properties

1. On your SAML IdP platform, navigate to the application you created for the Authentication Server.
2. Collect the following value:

   `IdP Metadata` – This metadata value is a base64-encoded IdP metadata document from your SAML provider.

Changing the Authentication Server to use SAML

```
 cd /opt/etxauthsvr
$ bin/etxauthsvr authtype enable saml
```

Edit the newly created `data/conf/saml/application.properties` file:

```
$ vi data/conf/saml/application.properties
```

Change the following value:

Configuration Field	Value
authsvc.saml.metadata	**IdP Metadata** This metadata value is a base64-encoded IdP metadata document from your SAML provider

Starting and testing the Authentication Server

1. Enter:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr start
```

2. In a browser, navigate to: `https://<server>:9443/osp/a/hc/auth/app/`

   Example:

   `https://<authentication-server>:9443/osp/a/hc/auth/app/`

   You will be redirected to your SAML Identity Provider for authentication.

3. You might be prompted to accept application permissions or approve the sign-in request. Once accepted, you should see the `Session Authenticated` message. This indicates that the Authentication Server has successfully authenticated using SAML.

> **Note**
>
> After you receive the `Session Authenticated` message, you can configure an ETX Site to use this Authentication Server, then test the login.

Exporting your Authentication Server settings to your ETX Site

Generate settings to be used by ETX Server to authentication with this ETX Authentication Server:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr export
ETX Auth Server URL: https://<authentication-server>:9443
ETX Auth Server Client id: hc-client
ETX Auth Server Client secret: hc-client-secret
Server Certificate PEM:
-----BEGIN CERTIFICATE-----
... public certificate data here...
-----END CERTIFICATE-----
```

1. Go to the **Site Settings -> Authentication page**.
2. In your ETX Site, log into the Server Manager.
3. Change the `Authentication Type` to `ETX Authentication Server`.
4. You can copy and paste the values from the `export` above into the fields on this page.
5. Press **TEST** to ensure communication to the Authentication Server is correct.
6. Press **SAVE**.

Changing the AuthType from the command line

You can configure the Authentication Server settings by running the `bin/etxauthsvr` script.

The Authentication Server only allows one active AuthType to be active at a time. The way to change this is by using the `authtype` command.

Determining the current AuthType

When the Authentication Server is first extracted, the authtype not configured. You can determine the current authtype by using the `config` command as follows:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr config authtype
authtype=
```

In this case, you can see that the authtype is blank so it is not configured. The Authentication Server will not start with an unconfigured authtype.

The AuthType command

The `authtype` command can be used to enable or disable authentication types. The Authentication Server only supports a single active authentication type at this time.

!!! note: If the authentication type needs to change, the Authentication Server must be restarted.

List of available semi-configured AuthTypes

Type	Description
ldap	Use an LDAP to authentication. The ETX Server itself supports LDAP but it is also supported through ETX Authentication Server
okta	Connect to OKTA for authentication.
entraid	Connect to Microsoft Entraid (Azure Active Directory) for authentication.
saml	Connect to a SAML/SAML2 provider.

Making an AuthType active

You can make an authtype active by using the `authtype` command, then confirming the change as follows:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr authtype enable okta
$ bin/etxauthsvr config authtype
authtype=okta
```

After the `authtype` command completes, you can see that the `config authtype` now displays `okta`.

Affects from activating an AuthType:

- The `data/conf/authtype` folder is created using the contents of `data/conf/templates/authtype`.
- The AuthServer uses the contents of this `data/conf/authtype` folder, especially the `application.properties` file, at start up to ensure that the Authentication Server uses the settings within.

Disabling an AuthType

You can disable an authtype by using the `authtype` command, then confirming the change as follows:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr authtype disable okta
$ bin/etxauthsvr config authtype
authtype=
```

After the `authtype` command completes, you can see that the `config authtype` now displays blank again.

> **Note**
>
> The `data/conf/authtype` folder is not affected by disabling an authtype. All customizations contained in tha folder remain.

Cleaning an AuthType

You can clean an authtype by using the `authtype` command, then confirming the change as follows:

```
$ cd /opt/etxauthsvr
$ bin/etxauthsvr authtype clean okta
$ bin/etxauthsvr config authtype
authtype=
```

After the `authtype` command completes, you can see that the `config authtype` now displays blank again. The `clean` argument first performs a disable on the authtype, then it cleans any customized files, restoring settings to default for that authtype.

When an AuthType is disabled, the `data/conf/authtype` folder is removed. Any customizations made to the `application.properties` file are lost.

> **Note**
>
> When a disabled authtype is enabled again, the contents from `data/conf/templates/authtype` are used to create the `data/conf/authtype` folder.

4. 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 ETX is to allow users to launch applications and desktops on a remote computer. When you launch a profile from ETX 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 ETX 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 ETX. 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 ETX Client once a session has been established. Also called a proxy host.

Dashboard

This web-based interface is your access point to ETX 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 ETX for each session, which records detailed information about the session for troubleshooting purposes.

etxscan utility

The ETX application scanner (`etxscan`) scans the connection node and returns a list of installed applications to ETX 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 ETX Server to launch those applications remotely. On Windows, etxscan also supports a `--syscheck` argument to print GPU hardware support.

ETX Client

The ETX Client consists of two parts - the 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

ETX profiles contain settings that define the look and behavior of ETX sessions. You launch sessions from profiles listed on ETX 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

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.

proxy host

See Connection Node.

published application

Applications installed on a Windows or X Window host can be published in ETX by means of an application scanner (etxscan) that is installed on the ETX 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 ETX window dynamically. The application (such as the KDE environment) will reflect this and fit in the new window. For example, you can 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 ETX, 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 ETX.

session

A connection to another computer, established by ETX, that moves information (including keyboard input and screens, for example) between them. You configure and launch these connections with ETX to work with one or more applications installed on the computer that you connect to. You can use ETX 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 ETX, 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 ETX 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.

5. Notices

Copyright

Trademarks

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

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