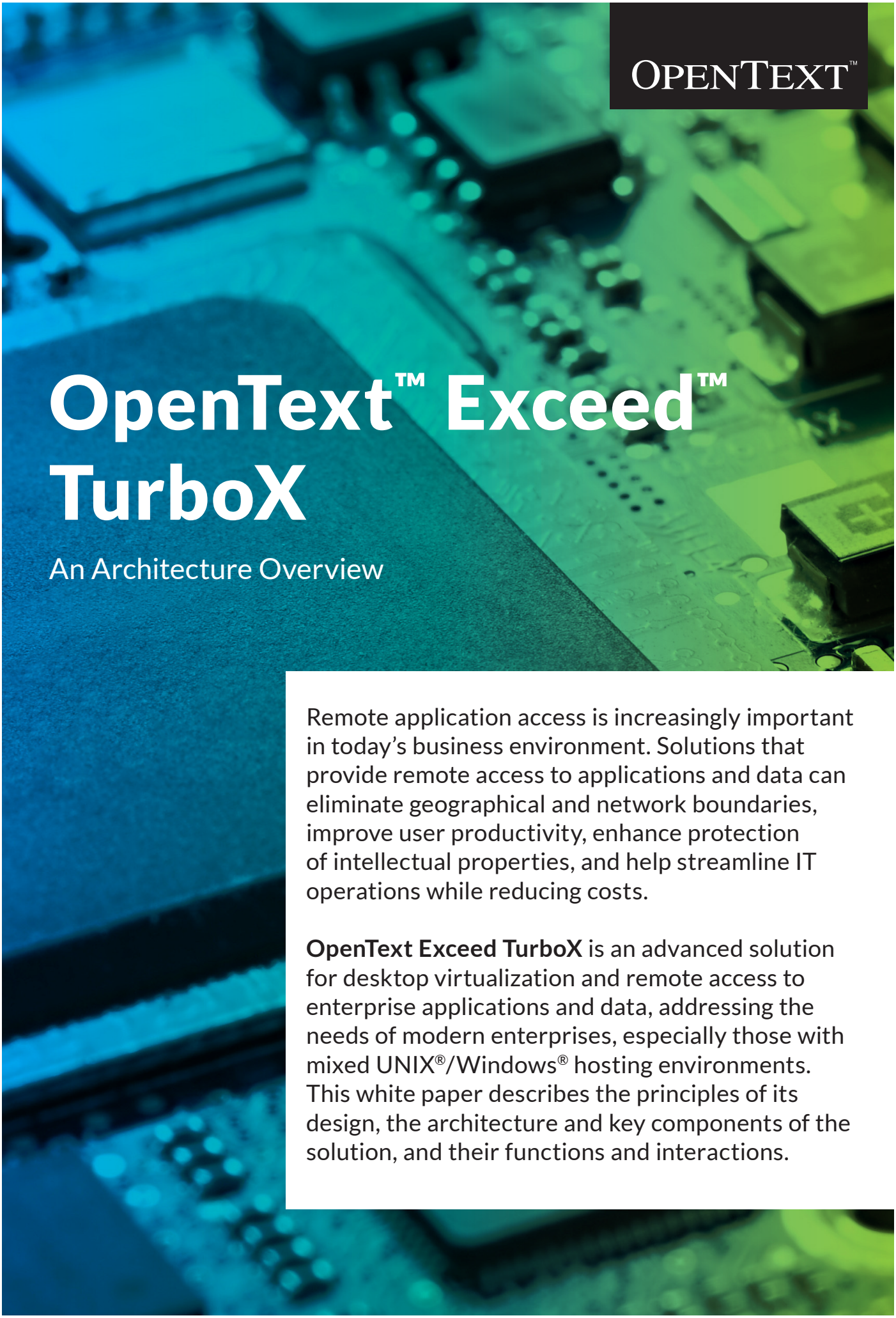


OpenText™

OpenText™ Exceed™ TurboX

An Architecture Overview

Remote application access is increasingly important in today's business environment. Solutions that provide remote access to applications and data can eliminate geographical and network boundaries, improve user productivity, enhance protection of intellectual properties, and help streamline IT operations while reducing costs.

OpenText Exceed TurboX is an advanced solution for desktop virtualization and remote access to enterprise applications and data, addressing the needs of modern enterprises, especially those with mixed UNIX®/Windows® hosting environments. This white paper describes the principles of its design, the architecture and key components of the solution, and their functions and interactions.

OpenText™

Table of Contents

1.0 Introduction

Remote application access has been a solution category since computers were first linked into a network. Businesses have always sought a solution that would allow users to access applications running on a remote host. Along came globalization, followed by all forms of virtualization, which pushed applications further away from users, while increasing the need for users to be productive while accessing applications and data from a great distance.

OpenText has been in the business of providing remote application access solutions for more than 20 years. Beginning with Exceed in the 1990s and including Exceed onDemand in the 2000s, OpenText has pushed the performance envelope, bringing complex 2D and 3D applications to enterprise users all over the world. OpenText Exceed TurboX is the future of remote application access solutions, made available today. It is a culmination of years of experience with X Window and network programming, and the understanding of hundreds of real world cases using business-critical UNIX and Microsoft® Windows applications.

Exceed TurboX, or ETX, is a complex solution that employs multiple tiers in its design. Components involved in this architecture are grouped into three categories:

- The user-facing components
- The administrator-facing components
- The Core X Server and Microsoft RDP Components

Six primary goals guided the conception and development of Exceed TurboX:

- Create streamlined and web-centric workflows for users and administrators
- Provide the fastest solution on the market in terms of text-based, 2D and 3D graphics-based performance
- Offer a consistent user experience across all supported client platforms
- Simplify the distribution and management of client software, profile and settings to support large deployment scenarios
- Offer fully auditable events for better visibility and accountability
- Work in tandem with various Desktop and Server virtualization strategies

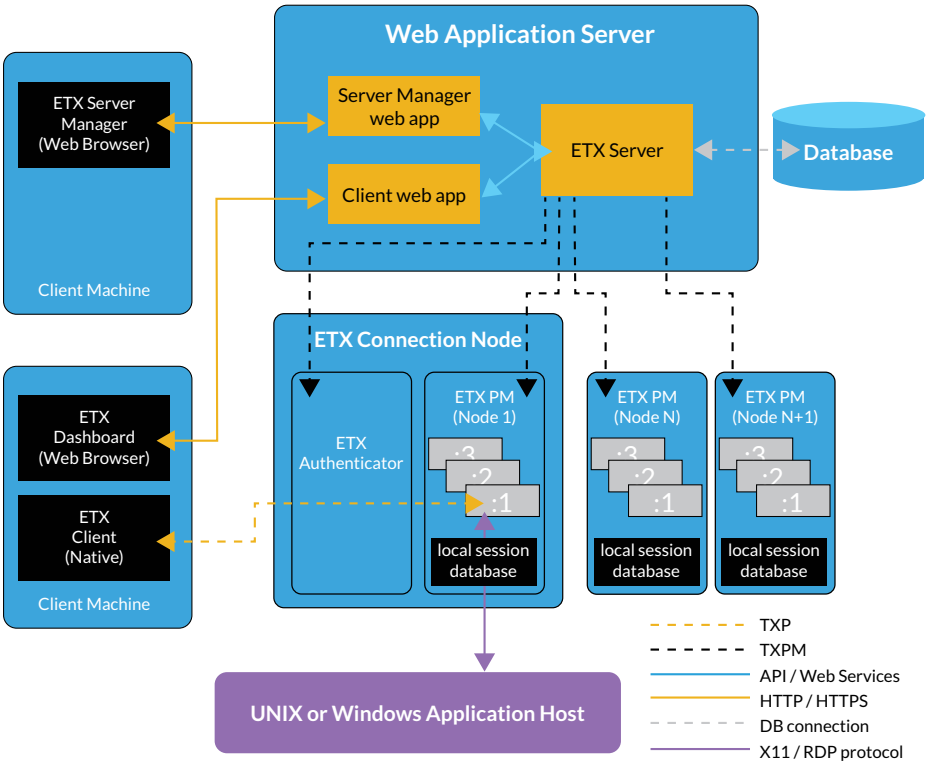
Exceed TurboX delivers results on all fronts. Its architecture is a departure from many traditional remote UNIX application-access solutions on the market, including our own Exceed onDemand. It relies on a hybrid of technologies to deliver high performance access with a great user experience and simplified administration.

2.0 The Architecture Diagram

The following diagram depicts the overall layout of all major Exceed TurboX components and their relationships. First, you will notice that ETX relies on a Web Application Server to provide a web-based graphical front end. Secondly, ETX uses a RDBMS in the backend.

They sandwich a group of ETX Connection Nodes, called the ETX Site.

We will further describe the functions of those building blocks that make up Exceed TurboX.



3.0 The Building Blocks

The following blocks of components are the foundation of Exceed TurboX.

3.1 ETX Server

This is the brain of the solution, functioning as concierge, gatekeeper, and traffic controller. It is a web application that runs on a Web Application Server and is presented to users when they first connect to the solution. ETX Server displays a web-based Dashboard to users or a Web Server Manager to administrators. Users can use the Web Dashboard to launch new sessions, create profiles, or manage their existing sessions. Administrators use the Web Server Manager to monitor and manage every aspect of the solution, from user enrollment to generating license and resource usage reports.

3.2 Web Application Server

Apache Tomcat™ is the Web Application Server that houses the ETX Server web application.

3.3 ETX Connection Nodes

Unlike ETX Server, an ETX Connection Node runs as a native application. With multiple nodes present in the infrastructure, ETX Connection Nodes can provide session load balancing and node failover.

ETX Connection Nodes are logical units, which contain three different sub-components, each handling a different task. During the installation of an ETX Connection Node, all three sub-components are installed: ETX Proxy Manager, ETX Authenticator, and Xstart Launcher.

3.3.1 ETX Proxy Manager

The main function of ETX Proxy Manager is to create remote session Proxies, manage and monitor the well-being of those Proxies, and communicate the status of the Proxies to ETX Server.

A Proxy is a fully functional X server/Windows RDP client, which supports all X protocol and RDP requests, and sends them in a compressed format to the ETX Client. Together, they are the only components in the solution that handle X or RDP application traffic. A Proxy supports all X protocol and RDP requests, and lets the Client handle screen drawing, and keyboard and mouse inputs. More information on the functions of Proxies and Clients is available in Section 6.0.

3.3.2 ETX Authenticator

As the name implies, this component handles user authentication. Depending on the type of authentication method chosen by the administrator during configuration, and the role that a Connection Node assumes, the Authenticator function may be enabled or disabled on a given ETX Connection Node. Note: Windows Connection Nodes cannot currently function as authenticators; at least one X Window node must be available to provide authentication capabilities.

3.3.2 Xstart Launcher

Xstart Launcher is the module responsible for launching individual X Window or Windows applications as specified in the ETX profile. ETX profiles include all configuration settings for the remote session, including which host to select and which applications to run.

3.3.3 Local Session Database

The local session database keeps track of session status and reports this back to ETX Server on a regular basis. If the connection between ETX Server and the Proxy manager fails, the local session database enables the Connection Node to continue operating normally, including suspending or terminating sessions. When the connection to ETX Server is re-established, the local session database is synced with ETX Server. This makes each node fault-tolerant in case of a network failure.

3.4 Database

The RDBMS is the bookkeeper of the solution. It's used to store all non-transient information, including server settings, user settings, profiles, event logs, license usage statistics, etc. ETX supports IBM DB2®, Microsoft SQL Server™, and Apache Derby, and comes bundled with Apache Derby for quick and easy installation. For load-balancing, fail-over, and high-availability capabilities, please refer to the database vendor for more information.

3.5 ETX License Server

Multiple ETX sites can share a single pool of licenses and have licenses dynamically or statically distributed to each site based on rules defined by administrators. The ETX License Server functionality is embedded in every ETX Server and can be enabled during setup. Once enabled, other ETX sites can connect to the License Server and use licenses from the pool, rather than using locally installed license keys. Special care has been given to the design of this solution to ensure that in case the License Server is out of service or out of reach, all sites will remain operational until the problem can be resolved.

3.6 ETX Client Runtime

ETX relies on a native client Runtime to handle remote communications and interactions. It is the counterpart to the ETX Proxy and communicates with the proxy continuously whenever a session is established. Like a Proxy, the client Runtime is a native application, which takes advantage of the best possible performance, reliability, and compatibility. Deployment of the ETX client Runtime is automatic; the correct Runtime is downloaded silently in the background whenever a user launches an application. Client Runtimes are available for Windows, Linux and Mac users.

3.7 Java Runtime Environment (JRE)

While it is not necessarily a building block on its own, JRE is a one of the system requirements. Naturally, JRE is required on the machine running the ETX Server and Apache Tomcat. JRE can also be used on the client's computer to download and execute the client Runtime when a profile is launched.

3.8 Native Client Launcher

The Native Client Launcher must be installed on the user's machine before the user can launch a profile in Exceed TurboX. The native client launcher will download and execute the native client Runtime in the background when the user launches a profile. The Launcher does not require administrator privileges to install, and it is provided for Windows, Linux®, and Mac® clients. If native installation is not allowed, the user can still launch sessions from any Java-enabled browser. The browser's Java plug-in will prompt the user to download and execute the ETX Client Runtime as an alternative deployment mechanism.

4.0 Connectivity

Between those building blocks, there are numerous network channels open to facilitate multiple facets of communications. The table below describes each type of network communication, its purpose, components involved, protocols used, and security levels.

DESCRIPTION	PRIMARY COMPONENT	SECONDARY COMPONENTS	PROTOCOL(S)	PORT(S)	SECURITY
TO BRING UP ETX WEB DASHBOARD	Web browser	Tomcat	HTTP/HTTPS	80/443 (default)	SSL
FOR ETX SERVER TO RETRIEVE DATA FROM DB2 AND DISPLAY IT IN THE WEB DASHBOARD	ETX Server	DB2 Database	JDBC	50000	Standard
FOR ETX SERVER TO RETRIEVE DATA FROM MICROSOFT SQL AND DISPLAY IT IN THE WEB DASHBOARD	ETX Server	MSSQL Database	JDBC	1433	Standard
WHEN A USER LAUNCHES A SESSION	ETX Server	ETX Connection Node	TXPM	15000 (default)	SSL
A SESSION IS PREPARED ON THE CONNECTION NODE	ETX Proxy Manager	ETX Proxy	TXPM	15000	SSL
THE NATIVE CLIENT CONNECTS TO THE PROXY	ETX Client	ETX Proxy	ThinX (TXP)	5510 (default)	SSL
SESSION DATA BEING RECORDED IN THE LOG	ETX Connection Node	ETX Server	HTTP (REST)	Same as ETX Server	None (SSL Possible)
LAUNCHING AN X APPLICATION	ETX Proxy	UNIX Application Host	X11	6000+	SSH
COMMUNICATION BETWEEN ETX SERVER AND LICENSE SERVER FOR LICENSE-RELATED INQUIRIES	ETX Server	ETX License Server	HTTP (REST)	Same as ETX Server	None (SSL Possible)

5.0 ThinX Protocol

ThinX Protocol is the digital blood coursing through the veins of ETX, and it is what makes ETX work so well, so fast. ThinX Protocol, or TXP, provides exceptional performance over any type of network condition, regardless of the available bandwidth or latency. Results of in-house analysis have shown that the amount of network traffic generated by TXP is less than one percent of the traditional X protocol. It is also more efficient than competitive remote access solutions.

Note: For Windows remote display, the connection between the Exceed TurboX proxy and the Windows host is RDP (3389), and the connection between the proxy and the client is ThinX. In this case, the proxy converts RDP protocol to ThinX protocol and sends it across the wire to the client desktop using TXP.

TXP is designed to accomplish the following:

1. Split functional tasks between the Proxy and Client
2. Reduce bandwidth requirements
3. Adapt to the network bandwidth
4. Reduce round-trip requests
5. Strengthen security

5.1 Split Functional Tasks Between Proxy and Client

The task of bringing an X application display to a user's desktop involves multiple network connections between different components at different phases of communication. TXP handles the communication link between the Proxy and the Client. A Proxy does not have the native ability to handle drawing requests and keyboard and mouse inputs. These X Window requests are mapped to the TXP-equivalent requests and passed on to the Client.

5.2 Reduce Bandwidth Requirements

TXP has the superior ability to compress, cache, and optimize requests. The goal is to minimize the amount of data that has to travel across the network between Proxy and Client, which is characteristically high in network latency and low in network bandwidth. Naturally, the work required to optimize and reduce network traffic comes at the expense of more CPU cycles and computing resources on both end-points of the connection. However, the abundant computing power that PCs and servers now harness can easily handle the increased workload without breaking a sweat, which allows Exceed TurboX to achieve bandwidth reduction without sacrificing overall performance.

5.3 Adapt to the Network Bandwidth

The level of compression and optimization can change automatically and dynamically depending on the real-time network condition – including bandwidth availability, latency, and even the nature of the business applications that are running. As the available bandwidth decreases, the response time of the server changes, or more graphically intensive applications are executed, Exceed TurboX will adjust and choose the right type of optimization to maintain performance and usability. The shift in TXP's behavior happens automatically and dynamically throughout the life cycle of the session without demanding user inputs.

5.4 Reduce Round-Trip Requests (X11)

X applications normally communicate with a Proxy over a LAN connection where network bandwidth is abundant. The Proxy, acting as a fully functional X server, is the gateway between X applications and the Client computing device. The Proxy is capable of monitoring and regulating the type and the amount of X Window commands that need to be passed to the Client over the typically slow network connection. Some X Window commands are interpreted and responded to by the Proxy, while other are batched up before transmitting to the Client in the form of TXP commands. With the intelligence built in to the Proxy, fewer round-trip requests will need to reach the Client, thereby eliminating inefficiency in the communication channel. The reduction in the number of round-trip requests over the WAN or Internet connection is significant, giving Exceed TurboX an unparalleled performance advantage.

5.5 Strengthen Security

TXP is secure in design. It can be easily encrypted using Secure Sockets Layer (SSL) protocol for heightened security. In fact, SSL-encryption is used by default to ensure business critical information is protected. Nothing is left to chance.

6.0 Client and Proxy Runtime

There are two administrative challenges that ETX is designed to solve:

1. The overhead associated with deploying and maintaining client software to every user's device
2. Updating the client software so that the installed client version and server version are kept in sync, without disrupting users or requiring ongoing deployment and upgrade efforts

We have arrived at an elegant solution for delivering the correct version of the client Runtime to the user, 100% of the time:

- Client and Proxy patches are developed and deployed in pairs. They are stored in the database as Runtime Packages. Runtime Packages are versioned, and within an application profile, a user can define which version of the Runtime Package should be used.
- When an application profile is selected for launching a new ETX session, ETX Server will read the profile and discover which version of Runtime Package the profile prescribes. ETX Server will then prepare the Runtime for use by the user and the Proxy Manager.
- On one hand, ETX Server will send the Client software through the Web Dashboard to the user's computer using the Native Client Launcher or Java Runtime. As long as the Native Client Launcher or Java applet are installed and accepted by users, the Runtime will be automatically downloaded and installed when a profile is launched. Installation of the Client Runtime and Native Client Launcher does not require elevated user privileges; instead the client software is installed in the user's personal folder.
- At the same time, ETX Server will send the other half of the Runtime Package – the ETX Proxy – to the Proxy Manager. Just like the Client software, the Proxy software is also designed to be a floating, separately deployable component, as opposed to a fixture in the infrastructure. The Proxy Manager is version-agnostic so it can manage any Proxy version.
- New Runtime Package can be deployed centrally using the ETX Server Manager web interface. Once a new version of Runtime Package is installed, the new version will be made available for users or administrators to use.
- To improve efficiency and performance, ETX Server may automatically distribute new Proxy software to Connection Nodes in advance.
- If the user's desktop or ETX Connection Node already has that version of software installed and available, there is no need to deploy any software, which can speed up the launch.

The above mechanism has the following advantages:

- Since Runtime Packages are developed as Client and Proxy pairs, compatibility is guaranteed.
- This approach also guarantees that the Client and Proxy are deployed at Runtime so users will never use a version of the Client that the Proxy cannot support.
- There is no need to manage desktop applications because ETX can deploy the right version of the Client software on demand without any administrative hand holding.
- If your organization has deployed a large number of ETX Connection Nodes, you don't need to constantly touch those server machines to install new Proxy software. The deployment of the Proxy is also handled by ETX autonomously.

The decision of which version of Runtime to use is prescribed in the profile – one can select a specific version of Runtime Package, or use the default Runtime package chosen by the Administrator. The simple act of tying a Runtime Package version to a profile offers a number of administrative flexibilities!

Imagine that tomorrow OpenText releases a new version of Runtime Package, you can deploy it centrally to the ETX Server, mark it as the default version, and every user and every profile will have access to it immediately! However, if you want to control the roll out of this new Runtime Package to a select group of users (for testing purposes, for example), simply create an application profile, which prescribes that specific version of Runtime Package, and assign the access right of that profile to a group of testers. These testers will run a different version of Client and Proxy software than the rest of the user community. Administrators do not have to reach out to those users specifically, install software on their desktop, or send out emails and reminders to teach them how to use the new version.

The best part is, Runtime Packages are designed to co-exist on the same machine. A user can run Profile A using the old Runtime Package, and simultaneously launch Profile B with the new Runtime Package. This type of flexibility allows administrators to eliminate any loss of productivity typically associated with software updates. Using the ETX Server Manager web dashboard, administrators can add new Runtime packages, set a Runtime package as the default, remove outdated Runtime packages, or put a Runtime package on hold so users will not be able to use it.

7.0 Authentication

ETX supports a full gamut of authentication methods:

- Lightweight Directory Access Protocol (LDAP)
- Microsoft Active Directory® (AD)
- Kerberos-based Single Sign On (SSO)
- Pluggable Authentication Module (PAM)
- Native User Credentials

Depending on your authentication requirements, ETX will use different components in the architecture to authenticate users.

AUTHENTICATION TYPE	COMPONENT THAT HANDLES AUTHENTICATION
LDAP, Microsoft Active Directory	ETX Server
PAM, Native User Credentials	ETX Connection Nodes

LDAP and Microsoft Active Directory are centralized identity management systems, which can be accessed from any computer. It is functionally simpler and more efficient to let ETX Server take care of those authentication types. All that is required is to properly configure LDAP settings in the ETX Server.

However, if your preferred authentication type is PAM or Native User Credentials, all authentication requests will be routed to one of the ETX Connection Nodes, which lets the component interact with the authentication mechanism of the underlying OS. That also means that you must assign the Authenticator role to at least one of your ETX Connection Nodes.

7.1 Node Roles

An ETX Connection Node can assume one of three roles: Authenticator, Proxy Manager, or both. You can assign the Authenticator or Authenticator and Proxy Manager role to more than one ETX Connection Node for high-availability purpose. However, you must make sure the PAM configuration or user database is identical across all Authenticators.

ROLES	FUNCTION
AUTHENTICATOR	Responsible for handling PAM and Native User Credentials authentication requests only
PROXY MANAGER	Responsible for creating and managing Proxies
AUTHENTICATOR & PROXY MANAGER	Both of the above

7.2 Making Changes to Authentication Type

Changing authentication type will have great impact on the functionality and availability of the solution. Choosing the authentication type, and subsequently changing it should be a well thought out and well-planned exercise. Because of the criticality of this feature, ETX Server must be stopped before changes can be made.

One of the potential impacts caused by changes of the authentication type after deployment is user account collisions. For example, user John Smith already has an account (jsmith) created by ETX Server; it's configured to use native authentication. The administrator then changes the authentication type to LDAP. Another user, James Smith logs on to ETX Server through LDAP using his LDAP username, also jsmith. James and John share the same username in two different authentication systems, but John's jsmith account already exists in the ETX database; so when James logs on to ETX Server, he will be accessing John's profiles, which is inconvenient at best and a security concern at worst. Therefore, before the authentication type is changed, administrators should carefully examine the user names in both authentication systems, identify potential username collision, and remove them from the database.

8.0 Clustering

8.1 What is a Cluster?

In a computer system, a cluster is defined as a group of machines that share common workloads and provide high availability (fallback) capabilities.. ETX has built-in support for clustering.

8.2 How Does Clustering Work?

If ETX were configured with only a single node to handle all user sessions, failure of that node (due to hardware, network or other failure) would be crippling. ETX would not be able to handle any new sessions until the node is restored and reconnected.

On the other hand, with two or more nodes present, failure of a single node does not cripple the system. The second node is able to take over the work load until the first node comes back online. All ETX Connection Nodes are registered to the same database used by ETX Server, and their presence is recorded and monitored so ETX Server knows the exact number of ETX Connection Nodes available, what each node is doing, and the amount of various resources available to each node. Together, they are connected and reported as a singular computing unit that users can access. From the user's perspective, there is no obvious difference between different Connection Nodes; in fact, most users should not care how many nodes are available and which node is handling the user's session. Just like when you access Amazon.com, you don't know, nor do you care, which one of a thousand different web servers is responding to your request.

Presented as a single logical unit, with display profiles and session in a unified Web Dashboard, ETX offers users a simple and elegant front end while hiding the scalable, powerful, and redundant backend architecture.

8.3 Multiple Authenticators

For organizations that choose to use native authentication or PAM, it is always a good idea to have more than one Authenticator in the system in case one of the ETX Connection Nodes is disconnected or otherwise unavailable. ETX Server will choose one of the available ETX Authenticators at random. Therefore, it is of utmost importance that the same set of users and their credentials are available or accessible by all ETX Authenticators in the system; otherwise some users may not be able to log in to ETX at any given moment.

In most cases, having two Authenticators in the system provides the necessary load balancing and protection against a single point of failure. More than two authenticators may be redundant and create unnecessary complication and workload involved in synchronizing native user credentials across multiple machines.

9.0 Licensing

9.1 Type of Licenses

ETX Licenses are concurrent in nature, meaning they allow a large number of users to share a smaller set of licenses. For example, if you bought 50 licenses for an ETX site, 100 users might share those licenses, so long as no more than 50 users have sessions active at the same time.

9.2 Rules of License Usage

The definition of a license in use is as follows:

- When a user starts the first X application session, a license is deducted from the license pool.
- Once the user takes possession of the license, that person can launch as many sessions from as many devices as he or she wishes.
- If a user shares a session with multiple users, then each user joining the shared session will consume one license (provided the user isn't consuming a license already).

9.3 Sharing Licenses Across Multiple ETX Sites

Multiple ETX sites can share a single pool of licenses and have those licenses dynamically or statically distributed to each site based on rules set by administrators. The ETX License Server functionality is embedded in every ETX Server, and can be turned on during setup. Once enabled, other ETX sites can be configured to connect to the License Server and use those licenses from the pool, as opposed to the locally installed licenses. Special care has been given to the design of this solution to ensure that in case the License Server is out of service or out of reach, all sites will remain operational.

9.3.1 License Server Basics

Any ETX Server can be configured as a License Server. During the installation, the administrator can define the role of the ETX Server to also be a License Server. It is also possible to make that switch after the setup, but the ETX Server must be first put into maintenance mode.

Conversely, an ETX Server can be configured at any time to use either locally installed license keys or connect to an existing ETX License Server. The switch between those two methods happens in real-time.

As part of our security model, when an ETX Server tries to join a License Server, the administrators of the ETX License Server will first need to validate the request.

9.3.2 License Distribution Methods

ETX License Server supports two types of license distribution: Static and Dynamic.

Static License Distribution

As the name implies, licenses are statically distributed to each ETX site so you can tightly control license usage and costs on a per-site basis. Once licenses are statically distributed to an ETX Site, the site will manage the license usage internally as if those licenses were locally installed.

Dynamic License Distribution

ETX Sites that have not been allotted a static number of licenses will share the remaining licenses in the license pool managed by the ETX License Server. As an ETX site requires a license, it will withdraw it from the License Server. Licenses can flow freely from one ETX site to another seamlessly.

However, even if multiple ETX Sites are sharing the same license pool, each is independently managing its license usage. If a user connects to two ETX Sites that are sharing the same license pool, this user will still consume one license from each ETX Site.

10.0 Fail-Over Scenarios

The ETX solution is designed to be resilient to various types of failures and offer continuous services. Disasters or failures can happen to computers hosting different components, and there are various degrees of failure. The table below outlines how each component handles each type of failure.

TYPES OF FAILURE		
COMPONENT	**NETWORK DISCONNECTION**	**PHYSICAL MACHINE SHUTDOWN OR DAMAGE**
CLIENT	If the machine where ETX Client is running is disconnected from the network, the user's session will be automatically put into suspended mode.	If the machine where ETX Client is running crashes or otherwise is powered off, the user's session will be automatically put into suspended mode.
PROXY MANAGER/ PROXIES	If the machine hosting the Proxy Manager and Proxies is disconnected from the network, naturally the Client will lose its connection to the session; but sessions will be put into suspend mode. However, other ETX Connection Nodes will con-tinue to accept new connections.	If the machine hosting the Proxy Manager and Proxies crashes or is powered down unexpectedly, user sessions will be permanently lost. However, other ETX Connection Nodes will continue to accept new connections.
AUTHENTICATOR	If the machine hosting the Authenticator is disconnected from the network, ETX Server will not be able to authenticate users. In this situation, no one will be allowed to connect to either the user Web Dashboard or the Web Server manager, with the exception of the maintenance user, which is built for the purpose of logging in to ETX to fix configuration problems with the authentication node. If the authenticator fails, it will not affect existing sessions; however new sessions will fail because they cannot authenticate the user.	
APACHE TOMCAT/ ETX SERVER	If Apache Tomcat or the ETX Server web app is unavailable – whether due to a broken network connection or a machine that has crashed or powered down – users will not be able to connect to the Web Dashboard. Consequently, they will not be able to interact with profiles or sessions, such as launching a new session or resuming a suspended session. Nor can administrators access the Web Server Manager. However, all active sessions will continue to operate without limitations. If a user suspends an active session while Apache Tomcat or ETX Server is down, he or she will have to wait until those systems are revived and online before the session can be resumed. Note: ETX supports high availability mode for the web server. This means that you can run two web servers in parallel. If the primary web server fails, the secondary will take over. If the backup server uses a separate database, the database should be configured to replicate the live database, otherwise information may be lost when switching to the backup server. Information about configuring a high availability environment for the web server and database is available as a whitepaper on OpenText Knowledge Center.	

DATABASE	If the database is unavailable because the database failed, ETX Web Dashboard and Web Server Manager will be inaccessible. However, all active sessions will continue to operate without limitations. When the database connection is restored, all functions will be resumed.	If the machine running the database crashes or is powered down unexpectedly, or the database is corrupted, ETX Web Dashboard and Web Server Manager will be inaccessible. However, all active sessions will continue to operate without limitations. When the database machine is restored all functions will be resumed.
	OpenText strongly advises companies to apply industry best practices when operating the database, and to make sure data is regularly replicated and proper high availability and fail-over protection is in place. Information about configuring a high availability environment for the server and database can be found in a whitepaper on OpenText Knowledge Center.	
LICENSE SERVER	**Static License Distribution** Upon disconnection from a License Server, an ETX Server will continue to manage the allotted licenses as if they were locally installed, for up to 72 hours. This provides time for IT to find and fix any failures. Impact to the ETX Server and its users is minimal unless this condition is left untreated. **Dynamic License Distribution** Upon disconnection from a License Server, each ETX Server will assume that any unallocated licenses are available for local sessions for up to 72 hours after the disconnection. When the connection to the License Server is restored, each ETX Server will report its current license usage to the License Server, and the License Server will recalculate the total number of licenses that are in use. Users who have active sessions will not be penalized if the number of licenses in use is greater than the number of licenses installed. In this case, this event will be logged and any user activities that require additional licenses will not be permitted until the total number of licenses in use is fewer than the number of licenses installed.	

11.0 Platforms

COMPONENT	PLATFORMS
CLIENT	Windows® 10 or later
	Windows® 8.1 or later
	Windows® 7 SP1 or later
	Windows Server® 2008 R2 SP1 or later *
	Windows Server® 2012 R2 or later *
	Red Hat® Enterprise Linux® 6.5 or later, 64-bit only
	SuSE Linux Enterprise Linux 11 or later, 64-bit only
	Mac OSX® 10.9 (Mavericks) or later
	(* Require Java Run Time 1.8 U60 or later)
ETX SERVER	Red Hat® Enterprise Linux® 6.5 or later, 64-bit only
	Oracle Solaris SPARC 10, 64-bit only
	Java Run Time 1.8 U60 or later
ETX CONNECTION NODE	Oracle Solaris SPARC 10, 64-bit only
	Oracle Solaris x86-64 10 or later, 64-bit only
	Red Hat Enterprise Linux 6.2 or later, 64-bit only
	SuSE Linux Enterprise 11 or later, 64-bit only
	IBM AIX 6.1 or 7.1
	Windows 10, 64-bit only
	Windows 8.1, 64-bit only
	Windows 7 SP1, 64-bit only
	Windows Server 2008 R2 SP1, 64-bit only
	Windows Server 2012 R2
DATABASE	IBM DB2® Express-C 10.5 with Red Hat Enterprise Linux 6 or later, 64-bit only
	IBM DB2 Enterprise Server 10.5 with Red Hat Enterprise Linux 6 or later, 64-bit only
	Microsoft SQL® 2012 SP1 Standard Edition or later on Windows Server 2012 R2 or later
	Microsoft SQL 2008 R2 SP2 Standard Edition or later on Windows Server 2008 R2 SP1 or later
	Apache DerbyDB 10 or later, on all supported ETX Server platforms
WEB BROWSERS	Microsoft Internet Explorer® 9 or later
	Microsoft Edge
	Firefox® 40 or later
	Chrome® 45 or later
	Safari® 8 or later (applicable to Mac OSX only)

Closing

Thank you for your interest in OpenText Exceed TurboX. To learn more about Exceed TurboX, visit our website at connectivity.opentext.com/etx.

About OpenText

OpenText is the global leader in Enterprise Information Management. Our software solutions help large enterprises manage and secure their unstructured digital data – the kind that doesn't easily fit into the rows and columns of a database – and use it to benefit their business and gain competitive advantage. To learn more about OpenText (NASDAQ: OTEX; TSX: OTC), please visit opentext.com.

Connect with us:

OpenText CEO Mark Barrenechea's blog

Twitter | LinkedIn | Facebook

www.opentext.com

NORTH AMERICA +800 499 6544 • UNITED STATES +1 847 267 9330 • GERMANY +49 89 4629-0
UNITED KINGDOM +44 (0) 1189 848 000 • AUSTRALIA +61 2 9026 3400