TGX

Administrator Guide

Version 2023.1

Mechdyne Corporation

April 2023

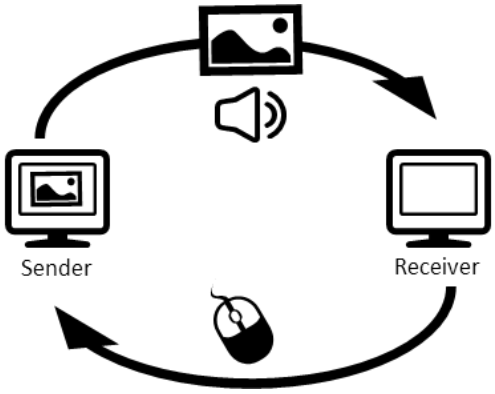
Mechdyne
ENABLING DISCOVERY

TGX ADMINISTRATOR GUIDE VERSION 2023.1

WELCOME TO TGX

TGX installers are available from the Mechdyne Software Portal. Contact Mechdyne Support by email to receive your login credentials and licenses for the software.

TGX provides separate installers for the sender and the receiver. The sender is the remote workstation whose desktop and applications are shared by TGX to a receiver. The receiver is the local computer that displays and interacts with the remote desktop of the sender via TGX.



TECHNICAL SUPPORT

Please submit questions and issues by email. A ticket will be created in the TGX support portal

EMAIL

software_support@mechdyne.com

LICENSE MANAGEMENT

TGX receivers require no license, however each TGX sender must have a license to accept a connection from a TGX receiver. The standard license model is based on the number of concurrent active TGX Sender sessions. This model allows TGX to be downloaded and installed on any number of sender and receiver machines, however the number of TGX sender licenses checked out at one time may not exceed the purchased concurrent count. If requested, Mechdyne will provide a nodelocked license for use on specific TGX senders.

FLOATING LICENSE

The floating license requires a license server to be setup using FlexLM. Please email the hostname, mac address, and OS of the VM or PC that will run the license server to software_support@mechdyne.com. The TGX support team will create the license and provide instructions on how to download and configure FlexLM license server software.

NODELOCKED LICENSE

A nodelocked license is based on the mac address of a TGX sender. To install a nodelocked license, the license file (e.g., `TGX.lic`) must be placed in a specific location. For Windows, the license location is `C:\ProgramData\Mechdyne\licenses`. For Linux, the license location is `/opt/mechdyne/licenses`.

SECURITY OVERVIEW

NETWORK TRAFFIC

TGX encrypts all network traffic with the OpenSSL implementation of TLS v1.2. The default cipher (configurable by the client's IT administrator) is ECDHE-RSA-AES256-GCM-SHA384, corresponding to Elliptic-curve Diffie-Hellman key exchange and AES 256-bit encryption of the channel. ECDHE defends against man-in-the-middle attacks and provides forward secrecy. The TGX installer will generate a 2048-bit RSA self-signed certificate/key pair on install or the client's IT administrator may choose to assign an enterprise certificate from a trusted root certificate authority.

ENTERPRISE USER AUTHENTICATION

Limiting access to remote workstations and servers by authorized users helps keep sensitive information safe. TGX supports local and domain account user authentication. On Windows, TGX can additionally be configured to only authorize users if they belong to the "Builtin\Remote Desktop Users" group.

To add users to this group, navigate to Windows Settings > System > Remote Desktop > User Accounts and choose the option to Select users that can remotely access this PC. Note users belonging to the "Administrators" group may connect via TGX without membership in "Builtin\Remote Desktop Users".

The config entry "RdpSidCheck=true" must be set under the heading [ServerSettings] on the Sender. For details see ConfigurationFileSettings below.

CONNECTION BROKER CREDENTIALS

TGX provides a mechanism to transfer credentials from a connection broker to a user's chosen TGX sender. In this scenario, TGX uses an AES 256-bit cipher and a random key to encrypt user credentials and creates a unique identifier for the credential-key pair.

The encrypted credential and identifier are transferred from the broker to the TGX sender, and the identifier and random key are transferred from the broker to the TGX receiver. The broker need not retain any part of the transferred data. The TGX receiver then provides the key and identifier to the sender at authorization time. The TGX sender automatically expires such credentials after 5 minutes (configurable) or upon successful TGX connection, whichever comes first.

Mechdyne provides a TGX encryption utility upon request to connection broker vendors or end clients creating their own broker.

VIDEO/AUDIO STREAMING

TGX communicates data by capturing the desktop image and audio generated on the sender and transmitting it to a receiver. The receiver then renders the sender's image directly onto the local screen and plays the audio on the receiver's sound system. By sending only the video/audio streams, TGX eliminates the need to share sensitive data over the network and also eliminates the need to have the application running locally. The data remains safe and protected on the sender, which can be in a secure facility/data center.

SENDER MONITOR BLANKING

If the sender workstation has monitors physically attached and a remote user starts a new session or the owner reconnects to an active session from a remote location, TGX will make a best effort to 'blank' the physical monitors. TGX will only blank monitors if the sender is running Windows and has an NVIDIA Quadro GPU. TGX does not attempt to blank monitors for other GPUs. The blanking operation prevents unknown persons from witnessing TGX sessions and keeps the desktop owner in control of sensitive data during the remote session. When the owner disconnects or logs off, TGX will return the desktop configuration to match that of the physical monitors and restore the blanked monitors.

COLLABORATION

The first user to login to a remote desktop is identified as the session owner. TGX allows other users and the session owner to collaborate, granting the session owner authorization and administrative privileges over their collaborators, to enhance security. Some examples of this include:

- The owner can grant or deny collaboration requests to prevent unknown/unwanted users from joining and seeing sensitive information.
- The owner can authorize collaborators to control mouse and keyboard input.
- Collaborators can never copy clipboard data back to their local computer. Only the desktop owner has the ability to copy clipboard data to his/her local machine.
- If the owner disconnects from the session, all other collaborators are also disconnected. This ensures collaborators cannot continue once the session has ended.

CERTIFICATE MANAGEMENT

TGX uses TLS v1.2 to encrypt communication between the sender and receiver. TGX can either use a user provided certificate/key or generate a self-signed certificate/key. It is recommended to use a full trust-chain verified certificate for maximum security. Using self-signed certificates may present additional dialogs to the connecting user, requiring them to accept a fingerprint similar to how SSH generally functions. When TGX encounters a self-signed certificate, the user is prompted to manually verify the certificate fingerprint (TGX does not perform hostname verification for self-signed certificates). The user can accept or decline the connection based on the fingerprint. The user also has the option to store that fingerprint to a *known hosts* file to skip the prompt in the future assuming the fingerprint does not change. If a conflicting fingerprint is encountered again for this host, the fingerprint conflict prompt will display.

When using a gateway, in order to avoid certificate validation errors, the same certificate should be used on all TGX senders served by the gateway. If a trusted CA certificate is to be used, the certificate host must match the gateway host.

The TGX Certificate Tool provides a command-line interface for generating/configuring certificates/keys and managing accepted fingerprints.

GENERATING CERTIFICATES/KEYS

The TGX Certificate Tool can generate key-pairs and self-signed certificates of varying key sizes to secure TGX communication.

WINDOWS

The certificate and key can either be stored in the system certificate store or as raw files. By default, the certificate tool places the certificate into the system certificate store, TGXCertificates.

```
tgx_certificate_tool --generate --key-size=<1024,2048,3072,4096,8192>
```

This command performs several operations:

1. Generates a self-signed certificate/private key pair with the specified key size.
2. Creates the TGXCertificates Certificate Store in the Local Machine System store if it doesn't already exist.
3. Stores the certificate/key to the TGXCertificates store.
4. Edits the TGX configuration file to appropriately point to the TGXCertificates store.

If TGX reports any errors while attempting to automatically store the certificate, please proceed to the section 'Manually Installing Certificates'.

To place the key and certificate into PEM files instead of in the certificate store, add the "--pem" option. This option places the private key in a plaintext file; we strongly recommend against its use in production systems. If using the PEM files, it is recommended that the key file be manually configured to only be readable by administrators.

Additional options when using the PEM flag:

```
--cert-path=<path to new certificate file>
--key-path=<path to new key file>
```

The cert-path and key-path options allow for specifications on where the tool will place the generated certificate/key files and their names.

LINUX

The certificate and key are stored as files. The generated key file will have restricted permissions.

```
tgx_certificate_tool --generate --key-size=<1024,2048,3072,4096,8192>
```

This command performs several operations:

1. Generates a self-signed certificate/private key pair with specified key size
2. Modifies the permissions on the key file to be appropriately restricting (600)
3. Edits the TGX configuration file to point to the generated certificate and key

Additional options:

```
--cert-path=<path to new certificate file>
--key-path=<path to new key file>
```

The cert-path and key-path options allow for specifications on where the tool will place the generated certificate/key files and their names.

USING EXISTING CERTIFICATES

The TGX Certificate Tool can also configure TGX to use existing signed certificates. The tool will also verify that TGX is able to use the certificate and key provided, otherwise, it will revert to the previous configuration.

WINDOWS

The certificate tool can load a PKCS#12/PFX file or PEM encoded certificate/key files directly into the certificate store (in most situations).

To load a PFX file to the certificate store:

```
tgx_certificate_tool --load --pkcs12 --cert-path=<path to certificate>
```

To load a PEM certificate/key to the certificate store:

```
tgx_certificate_tool --load --pem --cert-path=<path to certificate> --key-path=<path
to private key>
```

To use a PEM certificate and key file without storing them in the certificate store:

```
tgx_certificate_tool --load --pem --as-file --cert-path=<path to certificate> --key-
path=<path to private key>
```

LINUX

When providing an existing certificate, the key and certificate files must be in PEM format. The following command will properly configure the TGX configuration file to point to the certificate and key files.

```
tgx_certificate_tool --load --cert-path=<path to certificate> --key-path=<path to
private key>
```

MANUALLY INSTALLING CERTIFICATES TO THE CERTIFICATE STORE

WINDOWS

1. Create the TGXCertificates store in the Local Machine System certificate store.
    - `tgx_certificate_tool --create-store`
2. Generate a PKCS12/PFX file or Convert a PEM encoded certificate and key to the PKCS12/PFX format.
    - Generate:
        - `tgx_certificate_tool --generate --as-file --pkcs12 --key-size=<key size> [--output=<path to file>]`
    - Convert:
        - `tgx_certificate_tool --convert --cert-path=<path to PEM cert> --key-path=<path to PEM key> --output=<output PFX file>`
          Note: The TGX Certificate Tool only accepts PEM encoded certificate and key files for conversion to PFX.

          -or-

        - `openssl pkcs12 -export -inkey <private key> -in <certificate> -name <optional friendly name> -out <output file name.pfx>`
3. Import the certificate/key to the TGXCertificates store.
    1. Right-click on the PFX file and select *Install PFX.*
    2. Select "*Local Machine*" as the Store Location.
    3. Verify the file to import and select Next.
    4. Check the box "*Mark this key as exportable*." Do not enter a password. Select Next.
    5. Select the "*Place all certificate in the following store*" option.
    6. Use the browse button to select the "*TGXCertificates*" store.
    7. Select Next. Select Finish.
4. Edit the TGX configuration file to advise TGX that the certificate is in the TGXCertificates store.
    1. Open the TGX config file (C:\ProgramData\Mechdyne\TGX\config.ini).
    2. In the [EncryptionSettings] section (if it doesn't exist add it to the end of the file) add:

       ```
       CertificateInStore=true
       ```

MANUALLY INSTALLING CERTIFICATES AS FILES

WINDOWS

1. Generate a certificate/key or use existing PEM encoded certificate and key files. Note the key files must only be readable by Administrators.

   ```
   tgx_certificate_tool --generate --as-file --pem --key-size=<key size> [--key-path=<path to key> --cert-path=<path to cert>]
   ```

   NOTE: By default, this outputs two files, tgx.crt and tgx.key, the certificate and private key, respectively, to the tgx_certificate_tool's directory.

2. Edit the TGX configuration file to advise TGX that the certificate is in the TGXCertificates store.
    a. Open the TGX config file (C:\ProgramData\Mechdyne\TGX\config.ini).
    b. In the [EncryptionSettings] section (if it doesn't exist add it to the end of the file) add:

       ```
       KeyPath=<Path to key file>
       CertificatePath=<Path to certificate file>
       CertificatesInStore=false
       ```

NOTE: Be sure to escape any backslashes in the paths by adding another '\'. E.g. C:\\Program Files\\Mechdyne\\TGX Sender\\bin64\\tgx.crt

LINUX

To manually install certificates on Linux, perform the following steps:

1. Move the certificate a private key to a desired location.
2. Ensure the certificate and private key are in a PEM format.
3. Configure the private key permissions to be sufficiently restrictive (600).
    - `chmod 600 <private key>`
4. Edit the TGX configuration file to point to the certificate and key files.
    a. Open the TGX config file `/opt/mechdyne/TGX/etc/config.ini`
    b. In the [EncryptionSettings] section (if it doesn't exist add it to the end of the file) add:
    `KeyPath=<Path to key file>`
    `CertificatePath=<Path to certificate file>`

CIPHER SELECTION

TGX also allows configuration of the cipher suite that TGX will use. By default, TGX uses `ECDHE-RSA-AES256-GCM-SHA384`. If using a TGX generated certificate, only RSA based ciphers are allowed, since the TGX Certificate Tool generates RSA keys. If a non-RSA cipher suite is desired, keys will have to be manually generated by another tool.

To set the cipher suite:

1. Open the TGX config file (`C:\ProgramData\Mechdyne\TGX\config.ini` or `/opt/mechdyne/TGX/etc/config.ini`).
2. In the `[EncryptionSettings]` section add the following line:

    `AllowedCipherList=<Cipher list>`

See Appendix A for a list of supported ciphers.

FINGERPRINT MANAGEMENT

The TGX Certificate Tool also provides the ability to manage authorized fingerprints. The tool can add/remove fingerprints from the authorized hosts' configuration file. Note, fingerprint management only applies to the TGX receiver.

ADD AN AUTHORIZED FINGERPRINT

Receiver Only

To add a fingerprint for a specific host, to the authorized hosts' configuration file:

```
tgx_certificate_tool --add-fingerprint --hostname=<hostname> --
fingerprint=<fingerprint>

Example: tgx_certificate_tool --add-fingerprint --hostname=tgx.test.local --
fingerprint=a2:7b:6:aa:2f:d9:87:a8:50:c7:8a:ad:3e:4d:2a:4e:2d:5:21:c3
```

REMOVE AN AUTHORIZED FINGERPRINT

To remove a fingerprint for a specific host from the authorized hosts' configuration file:

```
tgx_certificate_tool --clear-fingerprint --hostname=<hostname>
```

To remove all fingerprints from the authorized hosts' configuration file:

```
tgx_certificate_tool --clear-all-fingerprints
```

MANUAL FINGERPRINT MANAGEMENT

The fingerprint cache is implemented as an ini configuration file, stored per user. The location of this file varies based on OS:

```
Windows: C:\Users\<user>\AppData\Roaming\Mechdyne\TGX.ini
Mac:     /Users/<user>/.config/Mechdyne/TGX.ini
Linux:   /home/<user>/.config/Mechdyne/TGX.ini
```

This file contains a list of hosts and their associated fingerprints in the format:

```
[hostname]
fingerprint=<fingerprint>
```

To remove a fingerprint, remove the `fingerprint=<fingerprint>` line for the desired host.

To add/replace a fingerprint, verify that the hostname does not already have an entry in the file. If it does, replace the fingerprint value with the new one. If it doesn't, add a new block at the end of the file in the format outlined above. The fingerprint must be formatted as ':' separated HEX bytes with any leading 0's removed. For example:

```
fingerprint=a2:7b:6:aa:2f:d9:87:a8:50:c7:8a:ad:3e:4d:2a:4e:2d:5:21:c3
```

To remove all fingerprint associations, delete the entire TGX.ini file. This file will be created automatically by TGX, if it doesn't already exist, the next time a fingerprint exception is required to be stored.

USB REMOTING COMPONENT

TGX provides support for USB devices being redirected from the receiver to the sender. This is currently only supported for human interface device (HID) class devices, though others may work. The USB component must be selected during install for both the sender and receiver to support this feature. This feature is only supported on Windows (sender and receiver). Support for USB redirection with macOS and Linux are scheduled for a future TGX release.

USB REDIRECTION CONFIGURATION

USB redirection must be selected during install on both the sender and the receiver. By default, only human interface device (HID) class devices can be redirected to the sender. The sender can be configured to whitelist/blacklist other classes or individual devices. A USB configuration file can be found in Appendix B.

The TGX USB configuration file can be found on the TGX sender in `C:\ProgramData\Mechdyne\TGX\usbConfig.ini`. The first section of the file, under the `[Class]` tag configures what USB classes are allowed to be redirected to this sender. Each following key corresponds to a USB class and can be true (enabled), or false (disabled).

The next two sections of the file, `[whitelist]` and `[blacklist]`, allow whitelisting and blacklisting specific devices by Vendor ID (VID) and Product ID (PID). Follow the Windows protocol for determining VID and PID of a specific device.

TGX includes a tool (Windows receivers Only) to help identify the USB Class Type, Vendor IDs, and Product IDs. You can find this tool on the TGX receiver in

C:\Program Files\Mechdyne\TGX Receiver\bin64\tgx_usb_tool.exe

Run the tool from a Command or PowerShell prompt.

```
…> C:\Program Files\Mechdyne\TGX Receiver\bin64> .\tgx_usb_tool.exe --print
Device Id  :: Class Name        :: Device Name
 46d:a56   :: Audio             :: Logitech H570e Stereo
138a:17    :: Vendor            :: Validity Sensors, Inc.
8087:7dc   :: Wireless          :: Intel Corp.
 4f2:b39a  :: Miscellaneous     :: Chicony Electronics Co., Ltd
```

The information printed by the tool is useful when setting up the USB Config on the TGX sender. The Class name tells you what class to enable to share that device. The Device Id shows what vid:pid to whitelist or blacklist.

AUDIT LOG

TGX will optionally log information on user connections/disconnections for selected Senders. The information includes:

- DateTime of an Event
- Hostname of the Sender
- Event type, one of:
    i. Connect –connection by owner, includes initial login and reconnection events
    ii. Disconnect – disconnection by owner, includes logout and disconnect events
    iii. CollabConnect – connection by collaborator,
    iv. CollabDisconnect – disconnection by collaborator
- Username of who is connecting to this Sender
- SessionId – internal assignment by operating system
- SenderUuid –unique identifier, use combination of SessionId and SenderUuid when trying to match a connection/disconnection for a user
- Server Version – TGX version running on Sender
- Receiver Hostname – computer name of the Receiver
- Receiver Address – hostname or ip address of Receiver
- Receiver Version – TGX version running on Receiver

In general, connect and disconnect events will be paired by Username and SenderUuid, for owner and collaborators. If there is a nonTGX connection, for example RDP, this information is not stored in the TGX audit log. If the RDP user disconnects, the event is not logged. If the same user, then connects by TGX, the log will show a CollabDisconnect by the user, then followed by a TGX connect event.

The TGX audit log is written to a folder that requires admin privileges to accesss. It is the responsibility of the Administrator responsible for the Sender to remove old log files.

The audit log is stored as a TSV file under:

(windows) C:\ProgramData\Mechdyne\TGX\audit

(linux) /opt/mechdyne/TGX/audit

By default, audit logging is disabled. To enable, add the following config entry under [ServerSettings]

EnableAuditLog=true

To allow audit logs to be included with a GetLogs operation, add the following config entry:

CollectAuditWithLogs=true

FULL COLOR AND CHROMA SUBSAMPLING

TGX starts sessions using chroma 4:2:0 subsampling. The benefit of subsampling is to reduce bandwidth requirements without significant image quality degradation. Artifacts from chroma subsampling are most obvious with text on a low contrast background or highly detailed visualization. To illustrate text issues with subsampling, we will use the image created by **RTINGS.com chroma test pattern** viewed on the Sender at 100% scale. The image shown below was captured on the TGX receiver using subsampling. Some of the lines are clear while others show degradation.

The image shown below was captured on the TGX receiver using full color. The lines that showed degradation with subsampling are now clear with full color. In addition, the color crispness is more evident on all lines.

A button on the TGX GUI is provided to allow the user to toggle between full color and subsampling on the fly. TGX uses AVC (H.264) encoding with chroma subsampling and HEVC (H.265) encoding with full color. The HEVC algorithm is more efficient and will keep the total bandwidth similar to what is created using AVC with chroma subsampling. If HEVC encoding is not available on the Sender GPU, TGX will not offer full color.

TGX will leverage GPU codec where possible on both the Sender and Receiver as shown in this table.

	Windows		Linux		Mac
	Sender	**Receiver**	**Sender**	**Receiver**	**Receiver**
AVC-420 codec					
NVIDIA	Yes	Yes	Yes	Yes	Yes
AMD	No	Yes	No	Yes	Yes
Intel	Yes	Yes	No	Yes	Yes
AppleM1	-	-	-	-	Yes
Software	Yes	Yes	Yes	Yes	Yes
HEVC-444 codec					
NVIDIA	Yes	Yes	Yes	Yes	No
AMD	NA	NA	NA	NA	NA
Intel	No	Yes	No	Yes	No
Apple M1	-	-	-	-	No
Software	No	Yes	No	Yes	No

NVIDIA 444 Encode requires Pascal or higher
NVIDIA 444 Decode requires Turing or higher
Intel 444 Decode requires iCore 11th generation or higher

CONFIGURATION FILE SETTINGS

TGX uses a configuration file which provides settings to better integrate with an enterprise IT deployment. There is a config.ini file on the sender and on the receiver and entries must be placed under the correct category in the file.

The categories include: [ServerSettings], [ClientSettings], and [DefaultSettings].

Each entry has a default value which will be used if there is not an entry in the config file. Each entry requires an argument of type: Boolean, Unsigned integer, or String.

To edit the config.ini file you must have **ADMIN** privileges.

NOTE: On the sender, the tgx-session-service must be "restarted" for the config.ini changes to be executed. Alternately, the sender can be rebooted. On the receiver, the TGX Launcher must be reopened for the changes to be executed.

The config files are stored:

Windows	C:\ProgramData\Mechdyne\TGX\config.ini
Linux	/opt/mechdyne/TGX/etc/config.ini
Mac	/Library/Application_Support/com.mechdyne.TGX/config.ini

[DefaultSettings] – Valid on both Sender and Receiver

CONFIGDEFAULT	*DESCRIPTION*
DefaultPort=40001	Defines the TCP port on the aender that the receiver will use to connect. This entry is automatically set during installation of TGX sender and receiver, but can be changed. The TCP port must be identical in the config.ini file on both the sender and the receiver or TGX will fail to connect.
Timeout=20000	Duration in milliseconds that receiver will wait for response from Sender that a connection is successful, otherwise the connection is considered a failure.

[ServerSettings] – Valid only on Sender

CONFIGDEFAULT	*DESCRIPTION*
AudioCaptureEnabled=true	If set to false, sender will not transmit audio to receiver.
DefaultAllowCollab=false	If set to true, sender will automatically accept collaborators without displaying a collaboration prompt to the owner. **SECURITY NOTE:** The owner will be unaware that a collaborator has joined their session unless they open the TGX Helper which will show all collaborators.
DefaultCollabPromptTimeout=15	Duration in seconds for the collaboration prompt on sender to be displayed before the default action is performed. There are two forms of default action: 1) If the collaborator is not the same user as the owner, the collaborator will be denied access; 2) If the collaborator is the same user as the owner, the current owner's session is disconnected and a new session is started for the user based on connection settings, e.g. display configuration, image quality, Etc.

DefaultCollaboratorInputEnabled = false	If set to true, the collaboration prompt sets the enable input checkbox to yes.
DisplayWhiteList = string (no default)	On physical workstation using X11 and a Nvidia GPU the display outputs may be presented by the driver in pairs (typically “DisplayPort” and “TMDS”). Only one type from this pair can be used at a time. If you run into issues with configuring displays, please refer to /var/log/Xorg.#.log to find the proper maximum output for your card(s) then add the outputs of the same type to the DisplayWhiteList. E.g., DisplayWhiteList = “DFP-0, DFP-2, DFP-4, DFP-6”
EnableDesktopConfiguration = true	This option is automatically set during installation of TGX sender, but can be modified anytime. If set to false, the sender desktop will not be reconfigured to match that of the receiver.
EnableDisplayBlanking = true	By default, TGX will blank the physical monitors attached to sender running Windows with NVIDIA Quadro GPUs. The monitors will continue to display the desktop if this option is set to false OR if the sender is using Linux or NVIDIA Geforce or non NVIDIA GPUs.
EnableUsb = false	This option is set during installation of sender based on whether USB is installed (true) or not (false). If USB is installed, this entry can be used to turn OFF USB. If USB is not installed, this entry cannot be used the turn USB on as sender would need to be reinstalled, selecting USB for install.
InboundClipboardSyncBytes=134217728	Maximum size in bytes of inbound clipboard (from receiver to sender), default is 128MB. A value of zero disables inbound clipboard sync.
MatchKeyboardLayout=true	If set to false, sender will ignore locale/keyboard settings of the receiver.
MatchKeyboardLayoutUsingPowershell = true	This is a new approach to setting keyboard layout on the sender. Set this value to false to continue to use the locale/keyboard approach. Note, this configuration has no impact if MatchKeyboardLayout=false, in which case the Sender’s locale/keyboard will continue to be used as is.
MaxCollaborators = 4	Maximum number of collaborators (including the session owner) allowed in a session. Set to 1 to disable collaboration. **NOTE:** TGX is not designed as a sharing tool for a large number of users, increasing this number could significantly reduce performance.
MultipleOwnerInputEnabled = false	If set to true, turns on the enable input checkbox on the collaboration prompt if the collaborator is the same user as the owner.
OutboundClipboardSyncBytes=134217728	Maximum size in bytes of outbound clipboard (from sender to receiver), default is 128MB. A value of zero disables outbound clipboard sync.
RdpSidCheck=false	If set to true, TGX will verify user is a member of the Remote Desktop Users group prior to granting them remote access to the system. **Windows ONLY**
UsbDeviceShareLimit=4	Maximum number of USB devices that can be shared to the TGX sender. Limit is 32.

[ClientSettings] – Valid only on Receiver

CONFIGDEFAULT	*DESCRIPTION*
AutoReconnectAttempts = 20	The number of attempts TGX Receiver will try to reconnect to Sender if network is lost. A value of 0 will disable auto reconnection.
CollapsibleMenuDistance = 5	Distance in pixels from the top of the screen for the mouse to trigger the display of the TGX GUI.
CollapsibleMenuHideTimeMs = 2500	Time in milliseconds that it takes for the menu to hide once it is shown. 2500 is 2.5 seconds.
CollapsibleMenuPlacement = Center	Align collapsible menu. Options are Center, Left, RIght
CollapsibleMenuShowTimeMs = 500	Time in milliseconds the mouse needs to be in the CollapsibleMenuDistance area for the menu to show up.
DefaultCaptureRate = 48	Target frame rate, value between 24 and 60 fps, default is 48.
DefaultClipboardSyncBytes=1048576	Default size of clipboard in bytes, that allows automatic synchronization of clipboard between sender/receiver. If the clipboard size exceeds this value, a popup message will be displayed to user and sync of clipboard is blocked. The default value of 1MB can be increased but not exceed MaxClipboardSyncBytes. There is a menu item in the receiver to "Enable Large Clipboards" which will ignore default clipboard size and auto sync up to the limit set by maximum clipboard size.
EnableShortcuts = false	Set to true to enable keyboard shortcuts.
EnableStatsLog=false	If set to true, enables menu item on receiver for user to start/stop recording of bandwidth and frame rate to csv file. These files are stored on the receiver with the naming convention “tgx_stats_XXX.csv” on: Windows C:\ProgramData\Mechdyne\TGX\logs Linux /tmp Mac /tmp
EnableUsb=false	This option is automatically set during installation of TGX receiver based on whether USB is installed (true) or not (false). If USB is installed, this entry can be used to turn OFF USB. If USB is not installed, this entry cannot be used the turn USB on as receiver would need to be reinstalled, selecting USB for install.
MatchKeyboardLayout=true	By default, receiver notifies the sender to synchronize with receiver's locale and set sender keyboard to the default keyboard defined for the locale. If set to false, the receiver does not provide info on locale/keyboard to the sender.
MaxClipboardSyncBytes = 33554432	Maximum clipboard size allowed for synchronization between sender/receiver when "Enable Large Clipboards" is set. The default value is 32MB. TGX limits this value to 128MB. NOTE: Large clipboards could take some time for complete transfer.
MiniMapHiddenAtStart=false	If set to true, the receiver will start with the MiniMap hidden and visibility of the MiniMap will be tied to the visibility of the receiver GUI.
MiniMapSnapDistance=5	Number of pixels from an edge that will trigger snap of viewing area to that edge

SuppressDisconnectPromptIfSessionManagerAllowsInteraction=false

If set to true, the Receiver will auto-disconnect with no prompt after the start of a timed logout/shutdown message from desktop manager. if false, TGX will prompt and wait for confirmation.

SuppressPopupOnSessionTermination = false

If set to true, don't display an error on the launcher when the remote session has been stopped through a logout or service stopped.

ACKNOWLEDGEMENTS

The TGX software package uses several third-party technologies, please see the User Guide for a complete description.

APPENDIX A: SUPPORTED CIPHERS

List of TGX supported ciphers:

ECDHE-ECDSA-AES256-GCM-SHA384
ECDHE-RSA-AES256-GCM-SHA384
DHE-RSA-AES256-GCM-SHA384
ECDHE-ECDSA-CHACHA20-POLY1305
ECDHE-RSA-CHACHA20-POLY1305
DHE-RSA-CHACHA20-POLY1305
ECDHE-ECDSA-AES128-GCM-SHA256
ECDHE-RSA-AES128-GCM-SHA256
DHE-RSA-AES128-GCM-SHA256
ECDHE-ECDSA-AES256-SHA384
ECDHE-RSA-AES256-SHA384
DHE-RSA-AES256-SHA256
ECDHE-ECDSA-AES128-SHA256
ECDHE-RSA-AES128-SHA256
DHE-RSA-AES128-SHA256
ECDHE-ECDSA-AES256-SHA
ECDHE-RSA-AES256-SHA
DHE-RSA-AES256-SHA
ECDHE-ECDSA-AES128-SHA
ECDHE-RSA-AES128-SHA
DHE-RSA-AES128-SHA
RSA-PSK-AES256-GCM-SHA384
DHE-PSK-AES256-GCM-SHA384
RSA-PSK-CHACHA20-POLY1305
DHE-PSK-CHACHA20-POLY1305
ECDHE-PSK-CHACHA20-POLY1305
AES256-GCM-SHA384
PSK-AES256-GCM-SHA384
PSK-CHACHA20-POLY1305
RSA-PSK-AES128-GCM-SHA256
DHE-PSK-AES128-GCM-SHA256
AES128-GCM-SHA256
PSK-AES128-GCM-SHA256
AES256-SHA256
AES128-SHA256
ECDHE-PSK-AES256-CBC-SHA384
ECDHE-PSK-AES256-CBC-SHA
SRP-RSA-AES-256-CBC-SHA
SRP-AES-256-CBC-SHA
RSA-PSK-AES256-CBC-SHA384
DHE-PSK-AES256-CBC-SHA384
RSA-PSK-AES256-CBC-SHA
DHE-PSK-AES256-CBC-SHA
AES256-SHA
PSK-AES256-CBC-SHA384
PSK-AES256-CBC-SHA
ECDHE-PSK-AES128-CBC-SHA256
ECDHE-PSK-AES128-CBC-SHA
SRP-RSA-AES-128-CBC-SHA
SRP-AES-128-CBC-SHA
RSA-PSK-AES128-CBC-SHA256
DHE-PSK-AES128-CBC-SHA256
RSA-PSK-AES128-CBC-SHA
DHE-PSK-AES128-CBC-SHA
AES128-SHA
PSK-AES128-CBC-SHA256
PSK-AES128-CBC-SHA

APPENDIX B: USB CONFIGURATION SAMPLE

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;  USB Base Class whitelist/blacklist
;
; The following section allows configuration specifying what classes of
;   USB devices are allowed/disallowed. If a class is not represented
;   it will be blacklisted (default: false). To whitelist a class
;   set the respective class key equal to true. To blacklist a class
;   set the respective class key equal to false. The settings in this
;   section can be overridden on a device by device basis using the
;   VID/PID whitelist/blacklist section below.
;
;   By default the only class allowed is HID, mouse/keyboard/etc.
;
; Example:
;   Hid=true ; whitelist all HID devices - except for ones in the VID/PID blacklist
;   MassStorage=false ; blacklist all MassStorage devices - except for ones in the
VID/PID whitelist
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
[Class]
Unspecified=false
Audio=false
Communications=false
Hid=true
Pid=false
Image=false
Printer=false
MassStorage=false
Hub=false
CdcData=false
SmartCard=false
Security=false
Video=false
Phdc=false
Av=false
Billboard=false
UsbCBridge=false
Diagnostic=false
Wireless=false
Miscellaneous=false
Application=false
Vendor=false

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; VID/PID Whitelist/Blacklist
;
; The following two sections allow for whitelisting and blacklisting
;   specific devices. This will override any class white/blacklisting
;   from above. To whitelist a device put its VID/PID in the Whitelist
;   section. To blacklist a device put its VID/PID in the Blacklist
;   section. The key for each VID/PID can be anything that helps
;   the user identify what the VID/PID represents.
```

```
;
; Device Formatting:
;   name=<VID>:<PID>
;
;  name - any unique identifier defined by user
;  VID - the Vendor ID of the device (in hex)
;  PID - the product ID of the device (in hex)
;
; Examples:
;   testWhitelistDevice=ABCD:1234
;   genericUsbMouse=1bcf:7
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;Whitelist specific VID/PIDs
[Whitelist]

;Blacklist specific VID/PIDs
[Blacklist]
```