TradeTek

Developer Guide

Version 1.0

Table of Contents

Click on an item to navigate to it.

Overview

TradeTek's framework includes a powerful yet easy to use API interface. The API interface allows you to read and write data to and from many areas within the application. It works by passing API commands through the clipboard either directly or through files. The API can either be accessed internally via a plugin or can be accessed externally from another application which has been granted access. The main function of the API is to read and write Job and Lists data using a tag to reference each object.

Getting Started

To get started working with the API, from the **File** drop-down menu click **Preference** to launch the Preference dialog. Check the **Developer** check box to activate the Developer tab. Check the **API** check box to turn on access to the API. The **Developer** tab should now be visible in the main tabs at the top of the application.

- **Caution:** The API commands are powerful and can overwrite data in several areas of the application or possibly corrupt data if not used correctly. We recommend creating a test job to use while experimenting with API commands.

Programming Interface

When the API is turned on, it polls every 0.25 seconds for commands to process. All external applications or plugins should send an API command, wait 0.25 seconds, and then start polling for a response every 0.25 seconds. Once a response is received and utilized, if the clipboard method is used the response should be cleared from the clipboard. If the file method is used, delete the TradeTekResponse file.

Syntax

- Every command is a string that contains the command name, open paren, zero or more required semi-colon separated parameters and closing paren.
- No unintended space character is allowed in the parameter list of a command.
- No quotes are used around literal text.
- Multiple commands can be chained together with | character for communication interface efficiency.
- If a parameter contains (,) or | character then enclose that parameter within {}.
- A command containing a parameter that uses {} convention cannot be chained.
- In general, never write these characters to a property value , ( ) | ~ ; #.

Developer Tab

From the **Developer** tab you can interact with the API by typing and sending commands in the **Command** field, then either clicking the **Send** button or tapping the **Enter** key. The response to the command will be displayed in the **Response** area. This is a great way to understand how all the API commands work and the data that they return. Type "Help" into the **Command** field and click the **Send** button to return a list of all the available API commands in the **Response** area.

The first step in creating a new plugin is to add its folder to the Plugin folder. From the **File** drop-down menu select **Preference** to launch the Preference dialog. Click the **Show** button to access the Preference folder. Inside of the TradeTek folder you will find the Plugins folder. Add a new folder inside of the Plugins folder for the new plugin, you can name it whatever you like. Place your executable, support files, PNG image file and optional PDF user guide in the new folder. For Windows the executable file name should be *Something*.exe, for Mac it should be *Something*.app. The base name of both executable files must be the same so that TradeTek can run the appropriate executable. Next, click the **Edit** button in the **Build Plugins and External Tools** button group in the **Developer** tab to create the new plugin.

The following buttons are in the **Build Plugins and External Tools** button group in the **Developer** tab.

- **Edit:** Launches the Plugin dialog where you can create or edit an internal plugin. Up to 10 plugins can be installed at one time. Click on the tab at the top of the dialog to select the plugin slot to add a new plugin to or select the appropriate plugin to edit. The Plugin dialog contains the following fields:
    - **Folder:** Click the **Select** button and choose your new plugin folder that was added to the **Plugins** folder.
    - **Version**: Set the version for this build of the plugin. Use a version number convention of X.Y where X and Y are one or two digit integers.
    - **Name**: Type the name of the plugin as you want it displayed in the **Plugin** tab (1 or 2 short words). The visible checkbox sets the plugin button in the **Plugin** tab to either visible or invisible.
    - **Executable**: Click the **Select** button and choose the executable file from your new plugin folder.
    - **Description**: Optionally enter information about your new plugin that will be displayed when the **Details** button is clicked in the **Plugin** tab. Use the left and right side **Description** area to enter 300 – 600 characters each.
    - **Vendor**: If you are going to make your plugin available to other users, optionally enter your company name, phone number, email address and website url.
    - **User Guide**: Optionally click the **Select** button and choose your user guide file from your new plugin folder (must have .pdf extension).
    - **Video**: Optionally enter a url to a video that can be played by the user by clicking the **Details** button in the **Plugin** tab (must have .mp4 extension).
    - **Authorization**: If the **Require Authorization to use API** checkbox has been checked in the **Protect API** dialog, you must add an authorization code for the plugin to run. If you plan on distributing the plugin to other users, be sure to add an authorization code. If you will only be using the plugin yourself and are not concerned about protecting the API, you can optionally leave this field blank.
    - **Ribbon Icon**: Click the **Select** button and choose an image to be used for new plugin. The image will be displayed in the **User Plugins** area of the **Plugin** tab. The image must be 46 pixels long x 46 pixels wide, 300 dpi and should be a PNG file. The image should also have a transparent background.
    - **Export Button**: Click to export the plugin into a file that can be imported by other users.

- **Protect:** This button launches the Protect API dialog. Check the **Require Authorization to use API** checkbox to protect against unauthorized access of the API. When checked, internal plugins are required to contain an authorization code to run in the application and external applications require a password to access the API. To add a password for an external application to connect to the API, click the **Add** button, enter a new password, and click **OK**. Multiple passwords can be added and passwords can be deleted as needed.
- **API Commands**: Opens a list of all API commands into a separate window that can be resized and moved around.
- **External Tool**: Launches a sample external application connected to the API. This will allow a developer that plans on creating an external application that will communicate with the API to understand how to setup the interface.

  There are two options for passing commands to and from the API, **Clipboard** and **File**. When **Clipboard** is selected, the **Send** button places the "TradeTekAPI=Command" string in the clipboard and polls for "TradeTekResponse=Response". When **File** is selected, the **Send** button writes the Command string for Windows into file C:\Users\Public\TradeTekAPI and polls for the Response string in file C:\Users\Public\TradeTekResponse. For Mac it writes the Command string into file /User/Shared/TradeTekAPI and polls for the Response string in file /User/Shared/TradeTekResponse.

  With the sample external application running, click the **Clipboard** radio button, type "Help" into the **Command** field and click the **Send** button. The actual string that gets copied into the clipboard is "TradeTekAPI=Help". The list of all API commands should be returned in the **Response** field. The actual string returned by the API response is "TradeTekResponse=*list of commands*". Now click the **Clear** button to clear the response, click the **File** radio button and click the **Send** button. The same list of all API commands should be returned in the **Response** field. This demonstrates an external application using both methods of passing API commands that allows the user to enter an API command and view its response.

- **Store Upload**: Uploading bundles and plugins to the store requires a developer account. To request a developer account, from the **Plugin** tab click **Developer** at the top left corner of the store page. Click the **Request Account** button, input your contact info and click the **Send Request** button. Once approved, your developer login credentials will be sent to the provided email address.

  Click the **Store Upload** button to launch the **Store Upload** dialog. Select the bundle or plugin that you want to upload from the **Product Name** drop-down list. All of the bundles and plugins that you have created should be available to choose from in the **Product Name** drop-down list. Enter your developer account **User Name** and **Password** and click the **Login** button.

  In the **Status** radio button area, if your bundle or plugin name is available to be used in the store the **Name Available** radio button will be set. To reserve your bundle or plugin name in the store, click the **Reserve** button. Once your bundle or plugin name has been reserved in the store, the **Name Reserved by You** radio button will be set. To set your bundle or plugin to the development phase, click the **Update** button. The **Status** radio button will change to **Development**.

  At this point you are ready to add a left and right side image for the store page and add your bundle or plugin file. Click the **Left Pic** button and select a PNG image that is approx. 300 pixels wide by 200 pixels tall. Next, click the **Right Pic** button and select the right side image of the same type and size. Click the **Product** button and select your bundle or plugin file. The upload process may take several minutes.

If you are interested in licensing and selling your bundle or plugin, from the **Plugin** tab click the **Developer** button at the top left corner of the store page and login with your developer account credentials. You should see the bundle or plugin that you just uploaded along with any other bundles or plugins that you may have previously uploaded. Click on the product that you just uploaded and fill out the licensing options and information. Click the **Request In Store Quote** button to receive a quote to set up the licensing for your product from Excel Software (licensing company). Once the license is set up, you will receive a ticket file and security code.

For licensed bundles, the ticket file and security code need to be added by clicking the **Build Bundle** button, selecting the ticket file and adding the security code, then clicking the **Build** button to re-build the bundle.

For licensed plugins, the ticket file and security code need to be referenced in a few API commands within your executable file. Once your code is updated, replace the old executable file in the plugin folder with the new one, then from the **Developer** tab click the **Edit** button and then click the **Export** button to generate the updated plugin file.

When you are ready to submit your bundle or plugin to the store for approval, click the **Ready** button. Once your bundle or plugin has been reviewed and approved, it should appear in the store within a week or so. Once added to the store, the **Status** radio button will change to **Active in Store** and your bundle or plugin will be listed in the store visible from the **Plugin** tab as well as the TradeTek Software website.

To update an existing bundle or plugin in the store, click the **Update** button which places the **Status** back to **Development** phase. At this point you can choose a new left and right side pic as well as an updated bundle or plugin file. Once you're ready to submit the update to the store, click the **Ready** button. Once your updated bundle or plugin has been reviewed and approved, it should appear in the store within a week or so. Once added to the store, the **Status** radio button will change to **Active in Store** and your bundle or plugin will be listed in the store visible from the **Plugin** tab as well as the TradeTek Software website.

To delete your bundle or plugin from the store, click the **Delete** button.

Use the link below to download a sample plugin called "Sort List". It's a plugin that sorts items in a list in TradeTek alphabetically. It includes the compiled executables for macOS and Windows and the PNG image used to build the plugin. It also contains the exported plugin, screenshot of the Plugins dialog, source code file and a ReadMe file that explains how it all works:

https://www.excelrt.cloud/TradeTek/help/Sort-List.zip

API Command Groups

The API commands are broken into groups. This section will provide a summary of what each group of commands is generally used for along with any pertinent additional information regarding specific commands.

- **Job:** These commands are used to perform general tasks related to jobs that have been created in the application. It is not required to have a job open to use these commands.
- **Job Property:** These commands are used to work with specific job properties and job property field names. A job must be open to use these commands.
- **Page:** These commands are used to work with specific page properties. A job must be open to use these commands.
- **Takeoff:** These commands are used to work with specific takeoff items or properties, including assembly sub-items. A job must be open to use these commands.
- **Takeoff Property:** These commands are used to work with specific takeoff properties and property field names. A job must be open to use these commands.
- **Subitem:** These commands are used to work with specific assembly subitems and properties. A job must be open to use these commands.
- **Subitem Property:** These commands are used to work with specific subitem properties and property field names. A job must be open to use these commands.
- **Folder:** These commands are used to work with specific Estimating tab folders and properties. A job must be open to use these commands.
- **Folder Property:** These commands are used to work with specific Estimating tab folder properties and property field names. A job must be open to use these commands.
- **List:** These commands are used to work with specific lists, list properties and reading and writing list data. It is not required to have a job open to use these commands.
- **General:** Below are some specifics regarding a few of the general API commands:
    - Refresh – Refreshes the screen after changes have been made. It's most efficient to make multiple changes and then refresh the screen at the end.
    - LastAPIWrite – Mostly used for support purposes. It timestamps when the last API write command was processed.
    - Authorize On/Off – These commands should be the first and last commands used. They authorize the commands to run by providing an authorization code and then close the API access once completed.
    - HttpGet/Post – Built in to simplify access to the internet. Mostly used to retrieve data from a web page or to fill in web forms.
    - TextFileRead/Write – Mostly used to save user preferences or share data between plugins.
- **License:** These commands are used for plugin licensing when the developer has chosen the **In Store** licensing option for selling and distributing their plugin(s).

Licensing & Selling Plugins & External Apps

Developers have a few different options for how they can license, sell and distribute both plugins and external applications that work with TradeTek. Once you have developed either a plugin or an external application that is ready for distribution, the next step is to create a TradeTek developer account. If you've created a plugin, be sure that you have imported the plugin file and that it's visible in the **User Plugins** button group in the **Plugin** tab. Click the **Details** button, click your plugin tab at the top and ensure that all the details look correct.

From the **Plugin** tab click **Developer** at the top left corner of the Plugin Store. Click the **Request Account** button, fill out the form and click the **Send Request** button. You will receive an email containing your new developer login credentials.

Once you've received your developer login credentials, from the **Developer** tab click the **Store Upload** button in the **Build Plugins and External Tools** button group. Enter your **Username** and **Password** and click the **Done** button to register your login information. From this point on, your login info will be saved and auto filled into the **Store Upload** dialog.

Click the **Store Upload** button again to launch the dialog and choose your plugin from the **Plugin Name** drop-down menu. If the plugin name is available, the **Name Available** radio button will be selected in the **Status** area. You can reserve your plugin name by clicking the **Reserve** button and you will notice the status change to **Name Reserved by You**.

While the plugin is in development phase, or if you need to update a plugin that has already been made active in the store click the **Update** button to refresh the plugin file. You will notice that the plugin status changes to **Development.** While in the development phase, from the **Plugin** tab click **Developer** at the top left corner of the plugin store and log in. You should see your new plugin listed under **Developer Plugins**.

Click on the plugin name to view plugin's details. You can choose to either distribute the plugin for free, sell the plugin through TradeTek's plugin store or sell the plugin from your own external store (website). If the plugin is going to be free, no further action is needed in the details area. If you will be licensing and selling the plugin through TradeTek's plugin store, click the **In Store** radio button and fill out the **In Store Purchase and Licensing Options** in the yellow area.

TradeTek works with a 3rd party software company called Excel Software to facilitate the licensing component of plugins and external applications. You can choose to provide multiple options for your plugin buyers. For example, Option 1 will license 1 computer for a monthly subscription cost of $19.95. Provide all the options that you would like to offer, fill in your contact info and click the **Request In Store Quote** button.

You will be contacted by Excel Software with a price quote for the 1-time setup cost and to discuss the required subscription to Safe Activation provided by Excel Software to handle the plugin licensing functionality. Safe Activation integrates with Stripe to handle payment processing. This creates a completely automated process so that when a plugin is sold in the store, the activation serial number is automatically sent to the customer and the payment goes directly into the developer's Stripe account.

When the plugin is ready to become active in the plugin store, click the **Left Pic** and **Right Pic** buttons and select .png files that are roughly 300 pixels wide by 200 pixels tall to be displayed on each side of your plugin's listing in the store. If the exact image size is used, no scaling is needed so the images will be crisper in the store. Check the **Windows** and/or **macOS** checkboxes to indicate which operating systems that your plugin supports. Next, click the **Plugin** button and select the file that was exported to the desktop when you created the plugin from the **Edit** button in the **Build Plugins and External Tools** button group located in the **Developer** tab. If you have misplaced the file, simply click the **Edit** button and then click the **Export** button to export the file to the desktop.

From the **Plugin** tab, click **Developer** at the top left corner of the plugin store and log in. Click the plugin name and then click **Preview** at the top left corner of your plugins area to see a preview of the plugin's entry in the plugin store. If everything in the preview looks good, you're now ready to submit the plugin to become active in the plugin store. Be sure to logout of your plugin area by clicking the **Logout** button at the bottom of the screen to return to the plugin store.

Click the **Store Upload** button and click the **Ready** button to change the status to **Ready for Store**. The submitted plugin will be reviewed by TradeTek. Once approved, the plugin status will be changed to **Active in Store** and the plugin will now be visible in the plugin store. TradeTek does not test your plugin at all. It simply reviews the Store entry page. The developer is solely responsible for all aspects of selling, licensing, and supporting their customers. If the developer later Updates an existing plugin, it goes back to Development status.

External applications can optionally be sold through the plugin store as well. A developer can create a "companion" plugin that can be submitted to the plugin store that allows the developer to license and sell external applications in the same way as plugins.