

Each command describes each TradeTek API command and the expected response. Each command consists of a text string that provides the command name and optionally open and close parens containing a list of semi-colon separated parameters. Each line below describes the name of a command, the parameters if any and a dash followed by a description of the expected response string.

TradeTek is a desktop application that runs on Mac or Windows. An executable using the TradeTek API can send a command and receive a response using the Clipboard or File interface.

To use the Clipboard interface, the executable places the string TradeTekResponse=BadCommand in the clipboard and polls for the response which is a string of the form TradeTekResponse=Response.

To use the File interface, the executable writes the Command string into one file on disk and polls to read the Response string from a different file. On Windows, the Command string is written into a text file at path C:\Users\Public\TradeTekAPI. The Response string is read from a text file at path C:\Users\Public\TradeTekResponse.

For Mac, the executable writes the Command string into file at path /User/Shared/TradeTekAPI and polls for the Response string in the file at path /User/Shared/TradeTekResponse.

- JobFolders - Return ; separated list of Job folders.
- JobDetails(JobFolder) - Return details for Job identified by Job folder.
- JobOpen(JobFolder) - Open job identified by Job folder.
- JobClose - Close open job
- JobSave - Save open job.
- JobStats - Count items in open job.
- JobName - Name of open job.
- JobDescription - Description of open job.
- JobNoteNames - Return ; separated list of Note sections for open job.
- JobNote(Name)- Value of named Note section of open job.
- JobLocation - Folder location of open job.
- JobPropertyGet(PropertyName) - Read property value of Job property identified by Property name.
- JobPropertyPut(PropertyName;PropertyValue) - Write property value of Job property identified by Property name.
- JobPropertyTags - Return ; separated list of Job Property tags.
- JobPropertyFieldGet(PropertyTag;FieldName) - Read field value of Job property identified by tag.
- JobPropertyFieldPut(PropertyTag;FieldName;FieldValue) - Write field value of Job property identified by tag.
- JobPropertyAdd - Add Job property and return tag.
- JobPropertyDelete(PropertyTag) - Delete Job property identified by tag.
- PageTags - Return ; separated list of Page tags.
- PagePropertyNames - Return ; separated list of Page property names.
- PagePropertyGet(PageTag;PropertyName) - Read property value of Page identified by tag.
- PagePropertyPut(PageTag;PropertyName;PropertyValue) - Write property value of Page identified by tag.
- PageImageGet(PageTag;Path;Mono) - Copy PNG image identified by Page tag from active Job to Path. If Mono is TRUE, copy B&W image otherwise color image
- PageImagePut(PageTag;Path;Mono) - Copy PNG image from Path to image in active Job for Page tag. If Mono TRUE, copy B&W otherwise color. See image change after reopen job.
- TakeoffTags - Return ; separated list of Takeoff tags.
- IntegrityCheck - Return SUCCESS or semicolon separated list of Takeoff or Takeoff.SubItems with a non-unique tag.
- TakeoffPropertyNames(TakeoffTag) - Return ; separated list of Takeoff property names.
- TakeoffPropertyGet(TakeoffTag;PropertyName) - Read property value of Takeoff identified by tag.
- TakeoffPropertyPut(TakeoffTag;PropertyName;PropertyValue) - Write property value of Takeoff identified by tag.
- TakeoffClone(TakeoffTag) - Return tag of cloned Takeoff specified by tag.
- TakeoffCloneNoMeasurements(TakeoffTag) - Return tag of cloned Takeoff specified by tag.
- TakeoffDelete(TakeoffTag) - Delete Takeoff specified by tag.
- TakeoffSubItemAdd(TakeoffTag) - Add SubItem to Takeoff specified by tag and return tag of new SubItem.
- TakeoffSubItemDelete(TakeoffTag;SubItemTag) - Delete SubItem in Takeoff specified by tags. Nested child SubItems are also deleted.
- TakeoffPropertyTags(TakeoffTag) - Return ; separated list of Property tags for a Takeoff identified by tag.
- TakeoffPropertyFieldGet(TakeoffTag;PropertyTag;FieldName) - Read field value of Property for Takeoff identified by tag.
- TakeoffPropertyFieldPut(TakeoffTag;PropertyTag;FieldName;FieldValue) - Write field value of Property for Takeoff identified by tag.
- TakeoffPropertyAdd(TakeoffTag) - Add Property of Takeoff identified by tag and return new Property tag. Use TakeoffPropertyFieldPut commands to modify property values.
- TakeoffPropertyDelete(TakeoffTag;PropertyTag) - Delete Property of Takeoff identified by tags.
- SubItemTags(TakeoffTag) - Return ; separated list of Takeoff SubItem tags based on provided Takeoff tag.
- SubItemPropertyNames(TakeoffTag;SubItemTag) - Return ; separated list of Takeoff SubItem property names based on provided Takeoff and SubItem tags.
- SubItemPropertyGet(TakeoffTag;SubItemTag;PropertyName) - Return property value from SubItem based on provided Takeoff and SubItem tags.
- SubItemPropertyPut(TakeoffTag;SubItemTag;PropertyName;PropertyValue) - Write property value of SubItem based on provided Takeoff and SubItem tags.
- SubItemSubItemAdd(TakeoffTag;SubItemTag) - Add SubItem to existing SubItem specified by tag and return tag of new SubItem.
- SubItemPropertyTags(TakeoffTag;SubItemTag) - Return ; separated list of Property tags for a SubItem of Takeoff identified by tags.
- SubItemPropertyFieldGet(TakeoffTag;SubItemTag;PropertyTag;FieldName) - Read field value of Property for SubItem of Takeoff identified by tags.
- SubItemPropertyFieldPut(TakeoffTag;SubItemTag;PropertyTag;FieldName;FieldValue) - Write field value of Property for SubItem of Takeoff identified by tags.
- SubItemPropertyAdd(TakeoffTag;SubItemTag) - Add Property of SubItem identified by tags and return new Property tag. Use SubItemPropertyFieldPut commands to modify property values.
- SubItemPropertyDelete(TakeoffTag;SubItemTag;PropertyTag) - Delete Property of SubItem identified by tags.
- FolderTags - Return ; separated list of Folder tags.
- FolderPropertyNames(FolderTag) - Return ; separated list of Property names for Folder specified by tag.
- FolderPropertyGet(FolderTag;PropertyName) - Read property value of Folder identified by tag.
- FolderPropertyPut(FolderTag;PropertyName;PropertyValue) - Write property value of Folder identified by tag.
- GlobalPropertyNames - Return ; separated list of Global Property names.
- GlobalPropertyValue(PropertyName) - Return value of Global Property specified by name.
- GlobalPropertyAssign(PropertyName;Value) - Assign value to named Global Property. Create property if it does not exist.
- GlobalPropertyDelete(PropertyName) - Delete Global Property identified by name.
- DefaultFolderTitleGet - Return default folder title.
- DefaultFolderTitlePut(Title) - Write default folder title.
- DefaultFolderPropertyTags - Return ; separated list of default folder property tags.
- DefaultFolderPropertyFieldGet(Tag) - Return default folder property Title and Value separated by ;.
- DefaultFolderPropertyFieldPut(Tag;Title;Value) - Write default folder property Title and Value for property identified by tag.
- DefaultFolderPropertyChangeTag(Tag;OldTag;NewTag) - Change default folder property tag.
- TemplateFolderTags - Return ; separated list of Folder tags.
- TemplateFolderPropertyNames(FolderTag) - Return ; separated list of Property names for Folder specified by tag.
- TemplateFolderPropertyGet(FolderTag;PropertyName) - Read property value of Folder identified by tag.
- TemplateFolderPropertyPut(FolderTag;PropertyName;PropertyValue) - Write property value of Folder identified by tag.
- FolderPropertyTags(FolderTag) - Return ; separated list of Property tags for Folder specified by tag.
- FolderPropertyFieldGet(FolderTag;PropertyTag;FieldName) - Read field value of Property of Folder identified by tags.
- FolderPropertyFieldPut(FolderTag;PropertyTag;FieldName;FieldValue) - Write field value of Property of Folder identified by tags.
- FolderPropertyAdd(FolderTag) - Add Property to Folder identified by tag and return Property tag.
- FolderPropertyDelete(FolderTag;PropertyTag) - Delete Folder property identified by tags.
- ListTags - Return ; separated list of List tags.
- ListPropertyNames - Return ; separated list of List property names.
- ListColumnNames - Return ; separated list of column names that can be used in a list.
- ListPropertyGet(ListTag;PropertyName) - Read property value of List identified by tag.
- ListPropertyPut(ListTag;PropertyName;PropertyValue) - Write property value of List identified by tag.
- ListDataGet(ListTag) - Read cell data for List identified by tag. Data starts with { and ends with } with columns separated by ; and rows by \.
- ListDataPut(ListTag;Data) - Write cell data for List identified by tag. Data format matches Get command with exact same column and row count.
- ListDataColumnGet(ListTag;Col) - Read column data for List identified by tag and column index. Returns list of column values separated by ;
- ListDataRowGet(ListTag;Row) - Read row data for List identified by tag and row index. Returns list of row values separated by ;
- ReportTags - Return ; separated list of Report tags.
- ReportTitles - Return ; separated list of Report titles.
- ReportPropertyGet(ReportTag) - Return ; separated property names Title;Columns;Rows;Active;FirstRow;LastRow.
- ReportDataGet(ReportTag) - Read cell data for active Report identified by tag. Data starts with { and ends with } with columns separated by ; and rows by \.
- ReportDataColumnGet(ReportTag;Col) - Read column data for active Report identified by tag and column index. Returns list of column values separated by ;
- ReportDataRowGet(ReportTag;Row) - Read row data for active Report identified by tag and row index. Returns list of row values separated by ;
- PropertyFieldNames - Return ; separated list of Property field names.
- Refresh - Refresh TradeTek screens.
- LastApiWrite - Display date of last API write command.
- AuthorizeOn(AuthorizationCode) - Provide authorization for one or more commands to run.
- AuthorizeOff - Unauthorize additional command streams to run.
- HttpGet(URL;Timeout) - Get data from URL. Abort if retrieval time exceeds timeout in seconds.
- HttpPost(URL;Data;Timeout) - Post formatted data to URL. Data of form Field1=Val1&Field2=Val2&FieldN=ValN. Return response from URL. Abort if retrieval time exceeds timeout in seconds.
- Execute(FilePath) - Run an executable.
- ShowURL(URL) - Present URL in default browser.
- TextFileRead(FileName) - Read a text file from the Plugin folder. FileName is cross platform, requires no path and returns nothing if file does not exist.
- TextFileWrite(FileName;Text) - Write text into named file in Plugin folder. FileName is cross platform, requires no path and overwrites file if it exists. Enclose text within optional {}.
- ExcelPath - Return path to Excel application or NoExcel if TradeTek cannot locate Excel.
- ExcelClose - Close Excel and return SUCCESS or NoExcel if TradeTek cannot locate Excel.
- TradeTekVersion - Return TradeTek Version.
- TradeTekBuild - Return TradeTek Build.
- TradeTekUser - Return User name assigned in TradeTek preferences.
- TradeTekOS - Return operating system running TradeTek of MacOS or WinOS.
- StringToHex(ascii text) - Return HEX representation of ascii text within the {}.
- HexToString(hex string) - Return string from HEX representation.
- Beep - When received, TradeTek will beep. This can be a useful test to ensure TradeTek is running with or without a GUI and has received the command before returning the Success response.
- Quit - Quit TradeTek. Put word TradeTekNoGUI in clipboard to launch TradeTek without a GUI. When an external App is done using the API, send Quit command.
- Help - Display list of API commands.
- LicenseCheck(Plugin;Ticket;Security) - Activate or validate a license based on the base name of the Plugin and Ticket and Security code for that Ticket.
- LicenseOptions(Plugin;Ticket;Security) - Present License Options dialog based on the base name of the Plugin and Ticket and Ticket and Security code for that Ticket.
- LicenseReset(Plugin;Ticket;Security) - Reset an activated license based on the base name of the Plugin and Ticket and Security code for that Ticket.
- LicenseLookupSN(Plugin;Ticket;Security) - Lookup recently purchased license Serial Number based on the base name of the Plugin and Ticket and Security code for that Ticket.
- LicensePaymentPortalURL(Plugin;Ticket;Security;OrderID;ButtonName) - Return URL of Stripe Credit Card portal using base name of the Plugin, Ticket and Ticket Security code. The OrderID and ButtonName fields refer to the Stripe/Safe Activation integration.
- LicenseVersion(Plugin;Ticket;Security) - Lookup the current version of this plugin to notify user if a newer version exists. Maintain newest version number in Notes field of Product page in Safe Activation using format Version=2.0.
- LicensePassthru(Cmd) - This command is used internally by TradeTek to direct send commands directly to the QuickLicense runtime.
- CloudLists - Return ; separated list of Cloud List names.
- CloudListTags - Return ; separated list of Cloud List tags.
- CloudListFieldGet(ListTag) - Read Name, Status and Users fields of Cloud List identified by tag. Response is semicolon separated fields.
- CloudListFieldPut(ListTag;Name;Status;Users) - Write Name, Status and Users fields to Cloud List identified by tag.
- CloudListFileGet(ListTag;Path) - Read CSV data for Cloud List identified by tag. Write CSV to disk at Path.
- CloudListFilePut(ListTag;Path) - Read CSV from disk at Path. Write CSV data to Cloud List identified by tag.
- CloudContacts - Return ; separated list of Cloud Contact names in the format First#Last
- CloudContactFieldGet(First#Last) - Read Cloud Contact data for First#Last entry. Response is First;Last;Company;Address;City;State;Country;Zip;OfficePh;CellPh;Email1;Email2;Notes;Users;Group
- CloudContactFieldPut(First#Last;First;Last;Company;Address;City;State;Country;Zip;OfficePh;CellPh;Email1;Email2;Notes;Users;Group) - Write Cloud Contract data for First#Last entry.
- CloudPreferences - Return ; separated list of Cloud Preference names.
- CloudPreferenceFieldGet(PreferenceName) - Return fields for Preference in the format Preference;Status;Users
- CloudPreferenceFieldPut(PreferenceName;Preference;Status;Users) - Write Cloud Preference data for PreferenceName entry.
- CloudJobs - Return ; separated list of Cloud Job names.
- CloudJobFieldGet(JobName) - Return fields for JobName in the format JobName;Status;User;Access;Group
- CloudJobFieldPut(JobName;JobName;Status;User;Access;Group) - Write Cloud Job data for JobName entry. CloudMessages - Return \ separated list of Cloud message. Each message contains Date, User and Message text separated by semicolons.
- CloudGenerals - Return ; separated list of Cloud General names.
- CloudGeneralFieldGet(GeneralName) - Return fields for GeneralName in the format General;Notes
- CloudGeneralFieldPut(GeneralName;GeneralName;Notes) - Write Cloud General data for GeneralName entry.
- CloudGeneralFileGet(GeneralName;Path) - Read General file identified by name. Write file to disk at Path.
- CloudGeneralFilePut(GeneralName;Path) - Read General file from disk at Path. Write file to Cloud General identified by name.
- CloudBudgets(JobName) - Return ; separated list of Cloud Budget names associated with JobName. The JobName and BudgetName pair are used to identify that Budget.
- CloudBudgetFieldGet(JobName;BudgetName) - Return fields for JobName;BudgetName pair in the format BudgetName;Notes;Users
- CloudBudgetFieldPut(JobName;BudgetName;BudgetName;Notes;Users) - Write Cloud Budget data for JobName;BudgetName entry.
- CloudBudgetFileGet(JobName;BudgetName;Path) - Read Budget file identified by JobName;BudgetName. Write file to disk at Path.
- CloudBudgetFilePut(JobName;BudgetName;Path) - Read Budget file from disk at Path. Write file to Cloud Budget identified by JobName;BudgetName.
- CloudInvoices(JobName) - Return ; separated list of Cloud Invoice names associated with JobName. The JobName and InvoiceName pair are used to identify that Invoice.
- CloudInvoiceFieldGet(JobName;InvoiceName) - Return fields for JobName;InvoiceName pair in the format InvoiceName;Notes;Users
- CloudInvoiceFieldPut(JobName;InvoiceName;InvoiceName;Notes;Users) - Write Cloud Invoice data for JobName;InvoiceName entry.
- CloudInvoiceFileGet(JobName;InvoiceName;Path) - Read Invoice file identified by JobName;InvoiceName. Write file to disk at Path.
- CloudInvoiceFilePut(JobName;InvoiceName;Path) - Read Invoice file from disk at Path. Write file to Cloud Invoice identified by JobName;InvoiceName.
- CloudPDFs(JobName) - Return ; separated list of Cloud PDF names associated with JobName. The JobName and PDFName pair are used to identify that PDF.
- CloudPDFFieldGet(JobName;PDFName) - Return fields for JobName;PDFName pair in the format PDFName;Notes
- CloudPDFFieldPut(JobName;PDFName;PDFName;Notes) - Write Cloud PDF data for JobName;PDFName entry.
- CloudPDFFileGet(JobName;PDFName;Path) - Read PDF file identified by JobName;PDFName. Write file to disk at Path.
- CloudPDFFilePut(JobName;PDFName;Path) - Read PDF file from disk at Path. Write file to Cloud PDF identified by JobName;PDFName.
- CloudJPGs(JobName) - Return ; separated list of Cloud JPG names associated with JobName. The JobName and JPGName pair are used to identify that JPG.
- CloudJPGFieldGet(JobName;JPGName) - Return fields for JobName;JPGName pair in the format JPGName;Notes
- CloudJPGFieldPut(JobName;JPGName;JPGName;Notes) - Write Cloud JPG data for JobName;JPGName entry.
- CloudJPGFileGet(JobName;JPGName;Path) - Read JPG file identified by JobName;JPGName. Write file to disk at Path.
- CloudJPGFilePut(JobName;JPGName;Path) - Read JPG file from disk at Path. Write file to Cloud JPG identified by JobName;JPGName.
- CloudCSVs(JobName) - Return ; separated list of Cloud CSV names associated with JobName. The JobName and CSVName pair are used to identify that CSV.
- CloudCSVFieldGet(JobName;CSVName) - Return fields for JobName;CSVName pair in the format CSVName;Notes
- CloudCSVFieldPut(JobName;CSVName;CSVName;Notes) - Write Cloud CSV data for JobName;CSVName entry.
- CloudCSVFileGet(JobName;CSVName;Path) - Read CSV file identified by JobName;CSVName. Write file to disk at Path.
- CloudCSVFilePut(JobName;CSVName;Path) - Read CSV file from disk at Path. Write file to Cloud CSV identified by JobName;CSVName.
- CloudEXCELS(JobName) - Return ; separated list of Cloud EXCEL names associated with JobName. The JobName and EXCELName pair are used to identify that EXCEL..
- CloudEXCELFieldGet(JobName;EXCELName) - Return fields for JobName;EXCELName pair in the format EXCELName;Notes
- CloudEXCELFieldPut(JobName;EXCELName;EXCELName;Notes) - Write Cloud EXCEL data for JobName;EXCELName entry.
- CloudEXCELFileGet(JobName;EXCELName;Path) - Read EXCEL file identified by JobName;EXCELName. Write file to disk at Path.
- CloudEXCELFilePut(JobName;EXCELName;Path) - Read EXCEL file from disk at Path. Write file to Cloud EXCEL identified by JobName;EXCELName.
- CloudFILEs(JobName) - Return ; separated list of Cloud FILE names associated with JobName. The JobName and FILEName pair are used to identify that FILE.
- CloudFILEFieldGet(JobName;FILENAME) - Return fields for JobName;FILENAME pair in the format FILEName;Notes
- CloudFILEFieldPut(JobName;FILENAME;FILENAME;Notes) - Write Cloud FILE data for JobName;FILENAME entry.
- CloudFILEFileGet(JobName;FILENAME;Path) - Read FILE file identified by JobName;FILENAME. Write file to disk at Path.
- CloudFILEFilePut(JobName;FILENAME;Path) - Read FILE file from disk at Path. Write file to Cloud FILE identified by JobName;FILENAME.
- MarkupCount - Return integer indicating number of markups in current job or NoJob.
- MarkupDataGet(Index) - Return string of ; separated Markup data fields at specified index.
- MarkupDataPut(Index;PageTag;MarkupType;Transparency;ShapeColor;LineWidth;PointSize;Selected;FontName;FontSize;FontColor;NoteText;Alignment;Link;Layer) - Change Markup at index with fields.
- MarkupAdd(PageTag;MarkupType;Transparency;ShapeColor;LineWidth;PointSize;Selected;FontName;FontSize;FontColor;NoteText;Alignment;Link;Layer) - Create Markup from fields.
- MarkupDelete(Index) - Delete Markup at specified Index.
- PropertyImageList - Return string of ; separated image tags from Property Image library. Each tag has form File-Name.png where File is a unique identifier and Name is optional image name.
- PropertyImageGet(Tag;Path) - Copy the image referenced by tag to the location specified by Path.
- PropertyImageAdd(Path;Name) - Add image at Path of type .png or .jpg to Property Image library and return uniquely assigned tag. Name parameters must be included but can be empty.
- PropertyImageDelete(Tag) - Delete image with specified tag from Property Image library.
- PatternImageList - Return string of ; separated image tags from Pattern Image library. Each tag has form File-Name.png where File is a unique identifier and Name is optional image name.
- PatternImageGet(Tag;Path) - Copy the image referenced by tag to the location specified by Path.
- PatternImageAdd(Path;Name) - Add image at Path of type .png or .jpg to Pattern Image library and return uniquely assigned tag. Name parameters must be included but can be empty.
- PatternImageDelete(Tag) - Delete image with specified tag from Pattern Image library.
- TextFromImage(ImagePath;TextPath) - Use default AI engine to extract text from PNG image at ImagePath and save to text file at TextPath.
- TextFromPDF(PdfPath;TextPath) - Extract text from PDF file at PdfPath and save to text file at TextPath.
- SearchJobText(Find;Range;TextPath) - Search previous captured Job text for Find string and save to text file at TextPath. Range of 0 is 1 line, 1 is 3 lines and 2 is entire file returned around Find string.
- AskAIAboutImage(ImagePath;Question;TextPath) - Use default AI engine to ask question about PNG image at ImagePath and save response to text file at TextPath.
- AskAIAboutPage(PageTag;Question;TextPath) - Use default AI engine to ask question about job page identified by tag and save response to text file at TextPath.
- AskAIAboutKB(KB;Question;TextPath) - Use default AI engine to ask question about Knowledge Base and save response to text file at TextPath. Use TradeTek, Job or custom name for KB parameter.