Clipjump v12.5

Home | About | Online Docs | Blog | Addons | Github

**Clipjump** is a multiple clipboard management utility for Windows. It was built to make working with multiple clipboards super fast and super easy. The program records changes in the system clipboard, stores them without any limits and provides innovative ways to work with them.

#### The basic ideas on which Clipjump is built upon

- No extra shortcuts needed to manage multiple clipboards in Clipjump, go with your default Ctrl + C,X and V.
- Not just the text. All Clipboard formats are supported.
- No opening a GUI and looking for your clip to paste it. Everything happens from Ctrl+v here, the pasting operation is superfast.
- Portability and Speed
- <u>Multilingual</u> and <u>Open Source</u>

The main idea of Clipjump is to make the clipboard management process fast as well as powerful. As a result, there are lots of features in here to suit your needs.

- Paste Formats Format text in different ways before pasting. The ones available by default include Trim Formatting, Sentence Case, Regex Replace, DeHTML and lots more.
- Plugins Using the <u>API</u>, you can create plugins that can very well extend the functionality of Clipjump.
- Channels You can sort clips in channels. These channels can be activated such that the clips copied automatically move to it.
- History Tool It stores a plain copy of each of the item copied onto Clipjump. By default, it stores the activity of last 30 days.
- Action Mode Another little innovation that saves you from remembring dozens of shortcuts for different features in Clipjump. Just press Ctrl ShiftA to see a list of popular functionailities and open them in a snap.
- Scripting <u>ClipjumpCustom.ini</u> is the script file for Clipjump. You can write small, easy-to-understand scripts to personalize Clipjump.
- Other features like adding tags to clips, exporting them, copying file

directly to clipboard etc are also supported.

#### **<u>Changes</u>** in this version

- Fixed crashing issues in v12.3 by using TooltipEx
- Now using SQlite database for History
- Ignore quick (<200ms) clipboard changes to prevent crashes
- [New] Option to use Win-C and Win-X for copying to Clipjump
- [New] Option to disable automatic clipboard monitoring

#### **DOWNLOAD**

### **Helpful Links**

### You may read about Clipjump at

- <u>PC Advisor</u>
- <u>GHacks Technology Blog</u>
- <u>MakeTech Easier</u>
- <u>BetaNews</u>
- <u>RgDot</u>
- FindmySoft (with video)
- <u>Softpedia</u>
- Giga (German)

Screenshots Online Documentation Translations Previous Clipjump page

# Help

### **First Steps**

- After Downloading Clipjump, unzip it and run **Clipjump.exe**. It's a portable application so no installation is needed.
- When you run the exe, Clipjump's icon will be seen in SystemTray.



- Try copying either by Ctrl + C or Shift + PrintScreen or by Right Mouse Click -> Copy
- You will notice a tip Transferred to Clipjump when you copy something. This will mean that the copy has been succeeded.
- Now, hold Ctrl and press V, you will see a tip containing the file or text to be pasted. If an Image has been copied, a Window showing your image will pop-up.



ttp://upsc.gov.in/exams/notifications/2014/scra/scra\_eng.pdf



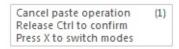
- Simply pasting the last copied item works as standard clipboard shortcut; release Ctrl to paste.
- To perform further actions, you will have to keep Ctrl pressed while releasing V only. This is **[Paste Mode]** 
  - To summarize: press Ctrl + v and then release V to enter Paste Mode

 To move through Multiple Clipboards, press v repeatedly while holding Ctrl.

Tap C while holding Ctrl to move backwards.

 To cancel paste operation, press x while in Paste mode. The tip should change to a Cancel Message. This is [Cancel Mode]

Release Ctrl to confirm.

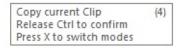


 To Delete Current Clipboard, press X twice in Paste Mode or once in Cancel Mode. The tip should now change to a Delete Message. This is [Delete Mode] Release Ctrl to confirm.

Delete current (2) Release Ctrl to confirm Press X to switch modes

To move or copy current clip to a different <u>channel</u>, press x 3-4 times while being in <u>Paste mode</u>. This is [Move Mode]-[Copy Mode]

Release Ctrl here to move or copy clips to a different <u>channel</u>. Detailed info about these features can be found <u>here</u>.



- To Delete all Clips, press x 5 times in Paste Mode (OR 4 times in Cancel Mode OR thrice in Delete Mode). The tip should change to a Delete all message. This is [Delete All Mode] Release Ctrl and then confirm the confirmation tip with Y to proceed.
- Use F while holding Ctrl in <u>Paste Mode</u> to search through clips. A search box is activated when you press F. Release Ctrl and type to filter clips containing the typed text. Use Up and Down keys to jump through the results, Enter to paste and Esc/Home to cancel. More in-depth info

of this feature can be found <u>here</u>.

### Notes

- As you can see, releasing Ctrl is the basis of performing actions in Clipjump.
- Also, note that **Cancel Mode**, **Delete Mode**, **Move Mode**, **Copy Mode** and **Delete All Mode** are inter-related. The program cycles through these modes when you press X while holding Ctrl. So, pressing X while holding Ctrl in **Delete All Mode** will switch back to **Cancel Mode**.
- TIP Press A in **Paste Mode** to directly jump to the first clip.
- TIP Press Q in Paste Mode to move a clip to first position.
- TIP Press F1 in Paste Mode to view <u>all the keys available in Paste</u> <u>Mode</u>.
- TIP Use keys 1....9 in Paste Mode to directly go to clip *num* ahead of current clip. For example if you are on Clip 7 of 25 within Paste Mode, 5 goes to Clip 12. Use the minus (-) key to subtract clip position and hence move back. Two times minus will make it plus again. Minus setting is made default to plus when Paste Mode is first opened.

### **Action Mode**

Action Mode provides a clean user-friendly interface to execute almost all the other components of Clipjump. Activated with the default configurable shortcut Ctrl + Shift + A, it provides <u>easy keys</u> to run components of Clipjump. Action Mode easy keys <u>can be totally configured</u> through <u>ClipjumpCustom</u> feature.

[v9.9.1] Action Mode is the default shortcut to execute any functionality in Clipjump. If you want to create System-Level shortcuts for these functions, you should use the <u>Settings Editor</u> OR <u>ClipjumpCustom</u>. **TIP** - Use keys 0 to 9 to change <u>channels</u> through Action Mode. Just press *channel number* when ActionMode popup is active. These keys can be overwritten through <u>ClipjumpCustom.ini</u>.

### Search in Paste mode (Search-Paste Mode)

[v10.6+] Press F to activate the Search feature in <u>Paste mode</u> (This is **Search-Paste Mode**). A box should come above the tip.

From here, you can search for specific text among all the clips in all the channels. Here are the key points-

- When Search box is active, release Ctrl and type your query . Don't worry, paste will not be invoked.
- As you type, you will see the tip jump to the first matched point. First it searches the currently active channel and then goes in ascending order to the next channels. Use Up and down to move Up and down the results
- Use Enter to paste if your result is in the underlying tip.
- Use Home/Esc to cancel paste operation.
- To later hide the box, hold Ctrl again and press F. Keep holding Ctrl now, you are in <u>Paste mode</u>.

The search follows the partial search methodology <u>as seen</u> in the History Tool.

#### **Important Info**

When using Search-Paste mode, the *clip pointer* will not be affected provided you end the activity with Search-Paste Mode. That is you use either Enter to paste or Home/Esc to cancel.

**Example** - The active clip will stay as it is (like Clip 4 of channel 0) even if you search something and that is in 5th clip of {6} channel. Use Enter to paste it and you will find that the active clip in Paste Mode is still clip 4 of channel {0} when you reopen it with Ctrl + V.

Clip pointer is not kept (updated) if you press Ctrl + F in Search-Paste Mode to close the search window. All other <u>Paste Mode</u> shortcuts like Ctrl + E for export, Ctrl + Space for fixate are still valid in Search-Paste Mode. While searching if you arrive at a tip you want to paste with a custom paste formatting, you can use Ctrl + Z and then Enter instead of Ctrl + F to close search window, then Ctrl + Z to a paste format and then release of Ctrl. The former method here will keep the clip pointer preserved **as it ends with an action of the Search-Paste Mode** and that's the <u>rule of thumb</u> to understand this topic.

**Tip** - You can change shortcuts in Search-Paste Mode through <u>ClipjumpCustom.ini</u>.

**Tip** - Search for *IMAGE* to find images.

**Tip** - You can make <u>Paste Mode</u> always start with search enabled from the <u>Settings Editor</u>.

#### Multipaste in Search-Paste Mode

You can use Ctrl + Enter when search is active in <u>Paste Mode</u> to paste without closing the paste and Search-Paste Mode. <u>Unlike Paste Mode</u>, here after first paste with Ctrl + Enter, the release Ctrl action will not change to cancel. So you don't need to hold Ctrl as you always do when [Search-Paste Mode] is active.

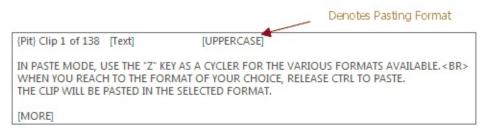
### **Paste Formats**

#### previously denoted by NO-Formatting Mode

[v10.7.3+] (Custom) Paste formats are possible as a plugin feature in Clipjump. With these, you can paste any clip as UPPERCASE, SentenceCase... or even trim off formatting from copied HTML.
To choose a format to paste with, you will have to activate it in the Paste Mode. As like many features, it is also available with a special key. In Paste Mode, use the z key as a cycler for the various formats available.
When you reach to the format of your choice, release Ctrl to paste. The clip will be pasted in the selected format.

Press Ctrl + V (activate Paste Mode) again. You will see that your selected format is still active. Use z to change it.

Note - To not remember last used formatting on Paste Mode restart, use the setting Always start paste mode with default formatting.



#### How it works?

All paste formats are powered by their respective plugins. Clipjump loads all the available paste formats from the plugin folder, creates a list and accesses them when user demands pasting/previewing from a plugin. You can too <u>create your own paste format</u> provided you know <u>AutoHotkey</u>, the simplest programming language.

Note that that Paste formats don't permanently change the clips. They are copied into memory and the 'memory-copy' is what is changed at the time of action.

**TIP** - To make a paste-format as default for Clipjump , see the option

Default Paste Format in the <u>Settings Editor</u>.

See Also - <u>Default Paste Formats distributed with Clipjump</u> See Also - <u>Video</u> demonstrating use of paste formats

# The Fixate Option

The Fixate option is meant to hold a current Clipboard at a fixed position while positions of all other Clipboards change.

Thus, if a Clipboard is fixed at [Clip 1], it will stay there at Clip 1 even after numerous Copies and Cuts.

This can be useful if you want a Clipboard entry to be easily accessible every time you go in <u>Paste Mode</u> using Ctrl + V.

Please note that Clipjump remembers last pasted clip's position. So, if you have just pasted [Clip 5], Clipjump will **start from [Clip 5]** the next time you press Ctrl + V.

Use the A in Paste Mode to go back to the first clip and thus change the clip pointer permanently.

#### **Using Fixate**

Pressing SPACE while in <u>Paste Mode</u> toggles the Fixed status of a Clip. If you reach the clip of your choice and want to fix it, press SPACE and you will notice a [FIXED] written after Clip x of y.

This means the clip is Fixed there. As said above, pressing SPACE again on the fixed clip will remove the FIX from it and the [FIXED] from the tip will disappear.



Note that the DeleteAll option will delete FIXED clips too.

### **Edit Clips in Paste Mode**

[v10.6+] Press H in <u>Paste Mode</u> to start editing a clip.

The Paste mode will end and your *default\_editor* or *default\_image\_editor* will open the clip. The *default\_editor* and *default\_image\_editor* can be configured from the *System* section of <u>Settings.ini</u>. Once the editor comes up, change the clip according to needs, save the changes and then **exit the editor**. Clipjump will respond to the exit and update the clip. Alternatively press Esc anytime to exit.

Editing works for all types of clips but for text clips all formatting will be lost when they are updated.

# **Adding Tags to Clips**

[v11.2+] Use the T key in <u>Paste Mode</u> to add tags to a clip. Tags show up in round brackets (tag) in the **Paste Mode** and they can be searched from the search in Paste Mode or the <u>Channel Organizer</u>. You can add tags to copied access-keys, passwords, shortened URLs....

Use the Properties option in <u>Channel Organizer</u> to add/edit tag.

# **Multiple Pasting in Paste Mode**

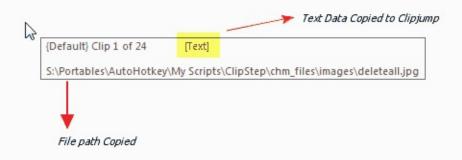
[v10.7.3] You can also use Enter in <u>Paste Mode</u> (while holding Ctrl) to do multi-paste. In this mode, you will be able to paste without closing the Paste Mode.

To do multi-pasting, tap Enter in Paste Mode. Once the first entry with Ctrl + Enter is pasted, releasing Ctrl will change action from 'pasting' to 'cancelling'. This change will be valid only in that session and once you have *cancelled*, everything will be back to normal i.e. *release* to paste and Ctrl + X and then release Ctrl to cancel.

The same feature is also available <u>when Search-Paste Mode is active</u>.

# **Copying Paths to Clipjump**

• Activating the <u>Action Mode</u> and then C will copy path(s) of selected files in all file managers including Windows Explorer. You can also create an indendent shortcut for this function in <u>Settings Editor</u>.



• Using <u>Action Mode</u> and then x will copy path of current opened folder in all file managers including Windows Explorer. You can also create an independent shortcut for this function in the <u>Settings Editor</u>.

# **Copying File content to Clipjump**

- Activating the <u>Action Mode</u> and then F will copy selected file's content to Clipjump.
- This function works with all file managers including Windows Explorer.
- This module will also work on Image files and <u>exported Clipboard files</u> (.cj).

# **Export and share Clipboards**

Clipjump supports exporting clipboards as files which can be stored as a record OR send to a friend.

The feature is accessible by tapping E in <u>Paste Mode</u> and by the shortcut Ctrl + E in the <u>History Tool</u>.

The lip or lipboard when exported will be saved in *My Documents* with a name export*x*.cj where *x* is a variable.

The .cj file extension when used with <u>Copy File Data</u> directs Clipjump to load the file contents into itself.

# **Copying and Moving Clip**

[v10+] Clips can be copied and moved from one channel to another from the <u>Paste Mode</u>.

To do so, press x when in Paste Mode to change to **Cancel Mode**. While still holding Ctrl, press x again and again till you reach to **Move clip** and **Copy clip** mode. Release Ctrl to confirm.

You will see a list of <u>channels</u> to copy/cut clip to. Select a channel and click OK.

# **Deleting clip after pasting (popping)**

[v12.3+] To automatically delete a clip right after pasting it, hold Shift along with Ctrl and then release both of them simultaneously. This will first paste the clip and then delete it. The active clip will be set to the clip right after the deleted clip.

# **Selective Windows Clipboard**

Many a times there are situations where you are not allowed to paste using the old shortcut Ctrl + V (like in Command Prompt). There the Selective Windows Clipboard Option comes handy.

#### Using

Cycle through the clipboards by Ctrl + V or C. When you reach the clip of your choice, press S while holding Ctrl.

You will see a Confirmation System Clipboard has Clip x in the form of a tip. That's it. Now, the Right-Click -> Paste or any other option except *Ctrl+V* will paste the clip you had choose like it is currently stored in system clipboard.

# **Copy Clipboard to Buffer (Hold Clip)**

[v10.9+] Hold Clip feature allows you to copy clipboard to a buffer location from where you can either paste the data or copy it to the currently active channel.

No shortcut for holdclip is assigned by default but the option has been provided in the <u>Settings Editor</u>. Once you have created a shortcut for holdclip, use it copy the selected text/file/folder/*anything* to the buffer. When copied, you will see a tip previewing the clip and all available options. Press F2 to open clip in <u>Common Formats GUI</u>, Ins to insert clip to channel and Ctrl+V to paste it.

**Note :** Pressing OK in Common formats GUI **exits** Hold Clip. So you can reuse Hold Clip when the output message box is shown.

Tip - Want HoldClip feature to cut rather than copy data to buffer. Try this <u>Customization</u> --

```
[cut_to_buffer]
bind = Ctrl + Win + X
STORE.holdClip_send = ^{vk58}
; virtual key 58 is X , so ^x = Ctrl+X
run = holdclip
```

# Sound when copied

[v9.8.1+] Clipjump can be configured to make system beep when some data is copied to it. The setting can be found in the <u>Settings Editor</u> as Beep when copied. To change the frequency of the beep played, navigate to the Advanced section of <u>Settings.ini</u> and change the value of *beepFrequency* to your liking.

## **Changing Paste-Mode shortcut**

To force Clipjump Paste Mode to be activated by some other key other than Ctrl + V, you can use the <u>Settings Editor</u> of Clipjump. Simply open the editor and look for the Paste Mode(Ctrl +..) value in the Shortcuts group. Change the default *V* in the box to some other key to make the Clipjump Paste Mode to be activated by the Ctrl-combination of that key. Make the value blank to disable Clipjump Paste Mode.

# **Disable Clipjump**

[v9.7+] You can disable Clipjump ( i.e. clipboard monitoring and <u>Paste</u> <u>Mode</u> ) from the Tray Menu as well as <u>Action Mode</u>. In Action Mode, this option is available with the easy key D.

# Auto disable Clipjump at certain windows

Clipjump can be temporarily automatically suspended when a certain window is active with this feature. The feature requires the active window's class which can be very easily managed with the Ignore Windows Manager located in the Tray Menu>Tools or in the Action Mode with easy key L. Use the Add Class button in the manager to start a tooltip application that shows the class of the active window. To make a window active, simply make it foreground by left-clicking on it.

Use Delete Class button to delete a value.

Note that you will have to restart Clipjump once you have made changes to the <u>Settings.ini</u> for them to take effect.

# **Resetting Clipjump**

[v9.5+] To reset Clipjump to its default settings and remove **all the data** created by Clipjump on your computer , use this option . The option can be found in the About window for Clipjump. The data that will be removed includes history items, <u>Paste Mode</u> clips, extracted icons and the <u>settings file</u>.

### **One Time Stop**

Many a times you have applications that use Clipboard's common transfer to extract data from various locations. These apps can be a headache when used with as consistent a tool as Clipjump.

The **One Time Stop** feature just solves this problem. Available with 0 key in Action Mode, you can call it once to make Clipjump automatically ignore the next data and linked content that is transferred to System Clipboard and so Clipjump. In this period, what really happens is that the Clipjump Clipboard Monitoring is deactivated.

**Note** - When Clipboard Monitoring is disabled (as in One Time Stop), Clipjump's tray icon will become dark.

Note - One Time Stop is deactivated when <u>Paste Mode</u> is accessed in the *one time stop activated* period.

## **Shortcut List**

\_

See <u>here</u>

<u>Translate</u>

# Channels

- <u>About</u>
- How to Use
- <u>Changing Channels in Paste Mode</u>
- Protected Channels
- <u>The "Pit" Channel</u> • <u>PitSwap</u>
- <u>Some Tips</u>
- <u>Channel Organizer (separate page)</u>

### About

Clipjump Channels are meant to group related clipboard items. So you can group all (copied) data about your ongoing project in one channel, office work in another channel and so on.. These channels have a name but are uniquely represented by their IDs. The default channel's id is 0 and is named as Default. You can use the <u>Channel Organizer</u> to manage your channel and its contents. (accessible from the <u>Action Mode</u> with easy key 0). To activate a channel, you can either use Channel Organizer or <u>Paste Mode</u>. We will see that later.

### How to Use

As said above, use the action mode shortcut or the Tray menu to open Channel Organizer window.

On the top-center region, you will see Active Channel list which will display the currently active channel. On a new installation, it should show Default as the channel 0 is active. Try clicking on that list and you will find that the list has only one element. This is because there is only one channel created by default.

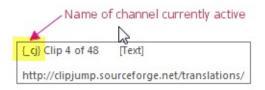
Suppose you want to create a new channel named "office". On the left side of the window, you will see a list corresponding to the list of channels. Right click on an item in the list (Default for example) and select New.

A box will appear asking the name of the channel. Write "office" and then click OK. You will see *office* has been added to the channel list. If you select it, you will find that the clips pane is empty meaning the channel has no clips. On the other hand, the Active Channel list is now showing *office* indicating that it is the currently active channel.

Another thing to notice here is that in the channel list, the number (**X**-office) before a channel's name is the ID given to it. ID starts from 0 and can go to infinity. Channels are ordered according to their ID's and not alphabetically. Now coming back, let's try copying something.

When you copy/cut data, you will see the confirmation tip Transferred to Clipjump to be preceded by the string {office} which will mean that the item you currently copied is going to office channel.

Now press Ctrl + v and release V to enter [Paste Mode] . You will see a {office} also preceding the Clip number message Clip x of y. As you have already guessed, this means that the clip is contained in the channel office and that it is currently active.



Note - You can use as many as channels you want and give them any name

you like.

#### IMPORTANT

Channel O Default is the main channel of Clipjump. It is the only channel for which the Minimum Number of Active Clips setting is valid, all other channel have no minimum and thus store unlimited clips. Also the setting Retain Clipboard data upon Application restart is invalid on channels other than O and so channels with (ID>O) store clips until they are deleted by the user.

### **Changing Channels in Paste Mode**

[v9.5+] You can change channels in the <u>Paste Mode</u> itself by using the Updown keys . Use the Up arrow key to increment the channel by 1 and use the Down arrow key to decrement the channel by 1. Changing channels via this method works like a cycle , so if you press the Up arrow key with the last Channel (say #2) active , you will be jumped back to the first channel (#0).

# **Protected Channels**

Protected Channels are user defined special channels that require confirmation before each copy of clipboard data to them.

If you copy some text while a protected channel is active, you will see a confirmation tip with a sharp beep to proceed further. Only after confirming that message, the clip will be added to that channel.

To create a protected channel, precede it's name with \_ i.e. a underscore . Any channel whose name starts with an underscore is a protected channel. **Tip** - To remove the beep, change the variable *protected\_DoBeep* to 0 from <u>ClipjumpCustom.ini</u>.

The code is protected\_DoBeep = 0. Add it in a section with no 'bind', one which is auto-executed at Clipjump start.

### The "Pit" Channel

The Pit channel is a reserved channel NAME with automatic Incognito feature. When a channel is named as Pit, the channel does not transfer it's clips to the Clipjump History thus creating a false Incognito mode situation. Note that when a channel named Pit is active, the Incognito mode is physically not turned ON and so you don't see the Clipjump's tray icon being grayed out.

The Pit channel can be quickly used to capture *junk* or *confidential* data to Clipjump and then delete them all at once with the <u>Delete All</u>.

### PitSwap

[v9.5+] PitSwap allows you to jump to an existing Pit channel from any other active channel with the ease of a hotkey. Activated with the <u>Action</u> <u>Mode</u> easy key P, it is a sort of 'channel toggler'. When pressing PitSwap shortcut when a channel such as Default(#0) is active, the channel will be automatically changed to a found <u>Pit</u> named channel. When pressing the shortcut again, the toggle will work in reverse changing the active channel back to the one you were using (here 'Default').

## **Tips for better productivity**

- For channel 0 to 9, you can also use <u>Action Mode</u> to change channels. See <u>How</u>.
- In Channel Organizer, don't forget to use Ctrl + G to quickly focus on the Active Channel list and then use arrow keys to change channel.

<u>Translate</u>

## **Channel Organizer**

#### [v11+]

<u>Channel</u> Organizer shows data stored in various channels of Clipjump and performs basic functions to manage them.

It can be activated from the <u>Action Mode</u> using the key 0 but the user can also create a system-level shortcut for it through the <u>Settings Editor</u>.

The left side list shows all the channels you have created with their channel numbers. The topmost item in the list i.e. the blank item shows clips of all the channels when it is selected.

Users can right-click on the clip to view options that are available. Many options have also been made accessible through buttons that are in between the two lists. The buttons do the common functions for both channels as well as clips and the **top horizontal arrow** indicates the direction for which other buttons will perform.

Options available like Insta-paste, edit clip, preview, search are all inspired from the <u>History Tool</u>.

You can edit clip's <u>tags</u> and <u>fixate</u> status using the **Properties option** in Context Menu. Make fixed = 1 in Properties to enable <u>FIXATE option</u> for a clip.

The **New Clip** button (Ctrl+N) can be used to create new clip in the currently selected channel. If all channels are selected i.e. the blank item, the clip is created in channel 0.

**The Delete option** currently doesn't support multi-selection so you will be available to delete only one item at a time.

## Preferences

• To make Channel Organizer show all clips i.e. choose no channel by default on startup, make the variable *ini\_OpenAllChByDef* equal to 1. You will have to use <u>ClipjumpCustom.ini</u> for the purpose. Just add the line ini\_OpenAllChByDef = 1 in a <u>non-auto-executing</u> section.

### **Global Window Shortcuts**

F5 - Refresh Ctrl+N - Create new clip Ctrl+Shift + N - Create new channel Ctrl+G - Set focus to Active Channel list Ctrl+F - Set focus to search box Alt+A - Set focus to Channel list (on the left) Alt+S - Set focus to Clips list

<u>Translate</u>

## **History Tool**

- Partial Searching
- <u>The Preview Window</u>
- Insta Paste
- Incognito Mode
- More Features
  - Edit Clip
  - Export Clip
  - <u>Hold Clip</u>
- <u>Disable History keeping</u>
- <u>Global Window Shortcuts</u>

## Introduction

Clipboard History tool stores record of all the clipboard items transfered to Clipjump in the last x days. It stores them as a listing in the order of recent first. The x days can be selected a/c your preference from the settings editor.

To open the Clipboard History tool , you can also use the H key in <u>Action</u> <u>Mode</u> other than the Tray Menu shortcut. A system level shortcut for the tool can be created from the Settings Editor.

To load full-size clip from the list provided by History tool, double-click on the item. To delete a history item , right-click on it and select Delete or select and press Del.

Tip - To view only Clipboard images in the History list, you can filter the listing something like [IMAGE] Note - Old users should note that the default system level shortcut Win+C was discontinued from v11.6.1

#### **Partial Searching**

When the checkbox 'Partial' is checked, the History Tool follows a partial checking pattern. Now you will be able to filter the Histroy List with *Space* delimited terms and only those clips will be shown which have all those space delimited terms in them.

#### The Preview Window

The History Preview Window previews the requested history clip. It supports preview of all types of clips from text to images. When previewing text , you have the option to search for terms (Ctrl+F) and the searching algorithm inhibits the Partial Searching setting from the History Tool. When previewing images, you can enlarge the image by resizing the preview windows . Maximize it to see the maximum possible size of the image preview.

#### Insta-Paste

Press Space or Middle Mouse Button after choosing your desired Clipboard item(s) in the history Gui to paste it directly to the underlying window. This method necessarily closes the History GUI. When selecting multiple items and then Insta-pasti'ng, the listing order is the order in which the clips are pasted.

**TIP** - To force Insta-Paste to write to Clipboard (Clipjump) when it is used, change *instapaste\_write\_clipboard* to 1 in the *[Advanced]* section of Settings.ini file.

**TIP** - To not close History Window on Insta-Paste, make *HisCloseOnInstaPaste* to 0 in the *[Advanced]* section of Settings.ini file.

#### **Incognito Mode**

When in Incognito mode, the Clipboard history will not be captured by Clipjump and the Clipjump icon in the System tray will fade (become greyyish). This feature can be very useful for maintaining Privacy. To enable Incognito mode, use the Tray menu.

#### **More Features**

Right Click on an item in the Clipboard History list to view more options like Export Clip, Edit Clip and Delete Clip(s).

#### **Edit Clip**

- With the Edit Clip option option, you can edit any clip with the <u>default</u> <u>Editor</u> or the default Image Editor. When the editor window shows up, you can make any changes and **save the file** (Ctrl+S). The clip will be reloaded in the History List once you have closed the editor. Note that this option works for images too.

#### **Export Clip**

- Exports the clip to My Documents with the name of *exportx.cj* where *x* is an incremented natural number. <u>The same feature</u> is also available for channel clips through **Paste Mode**. An exported clip can then be loaded back into Clipjump using the <u>Copy File Data</u> feature.

#### Hold Clip

[v11.6+] Hold Clip copies the selected clip to buffer and then allows you to paste it in any window without disturbing the Clipjump clipboards. <u>The same feature</u> is also available globally.

#### **Disabling the History tool**

- Change the Number of days to keep items in History setting to 0 to permanently disable History tool. This will make Clipjump automatically start with the **Incognito Mode** on as a symbol to show that History is not captured.

Please note that disabling the history tool in the way as said above will not delete old history items unlike to the fact that if you change the Number of days to keep items in History from 50 to 2, the history older than 2 days will be deleted by Clipjump.

### **Global Window Shortcuts**

1

F5 - Refresh Ctrl+F - Focus on Search box

<u>Translate</u>

## **Common Formats Plugin**

version 0.61

## Introduction

**Common Formats** is a pformat plugin that lets you paste the clip with various inbuilt paste formatting. Users can add more formats in the *pformat.commonformats.lib/user.ahk* file.

Note - This file is not present by default. It is created when the Common Formats plugin runs for the first time ( either from the plugin manager or paste mode ).

## Using

Once you have selected the **Common Formats** paste-format, releasing Ctrl to paste will bring up the GUI.

Use Up or Down keys or mouse to select your needed format and press Enter or OK button to paste.

You can press Esc or simply close the GUI to cancel the paste operation. Selecting the paste-format **None** will paste the original clip without making any changes.

**The Apply button** applies the current changes made the formatting to the clip to be pasted. Thus it helps you in applying multiple formats to a clip in the needed order. Note that the None paste format will still paste the real original clip and the 'apply' won't be undertaken.

**Input Field** is used by formats that have parameters in addition to the input text. RegExReplace for example uses input field to get the other 2 parameters, the match needle and the replacement text and the users are supposed to enter these two parameters one in a line (line by line).

## List of Formattings available

BBCodeList DeHTML HTMLList lowercase TrimFormatting NumberedList **RegexReplace** TrimWhiteSpace UnHTML UPPERCASE

# Porting an existing pformat to under Common Formats

To make any independent pformat a part of the **Common Formats**, you can simply copy its code to the *pformat.commonformats.lib/user.ahk* file. Then replace the main function name plugin\_pformat\_name by plugin\_pformat\_commonformats\_name where name is the name of that independent pformat. Also it will be a good idea to study the code of *pformat.commonformats.ahk* first.

You can also create a new format under the **Common Formats** user file. It should be easy provided you know <u>AutoHotkey</u>. Just make sure you name the new format function as plugin\_pformat\_commonformats\_*MYNAME* where *MYNAME* is the unique name of your format and this is what that will appear in the GUI. (Related: pformat)

**Info Data** for the format shown in the respective field is nothing but data stored in STORE object as STORE["commonformats\_" formatName] where formatName is the name of the format such as RegexReplace.

<u>Translate</u>

## **Settings Editor**

## General

Clipjump Settings Editor provides a visual interface to change the basic settings of Clipjump.

The Settings Editor is located in Tray Menu and can be accessed via Tray Menu>Tools>Settings or via <u>Action Mode</u> easy key E (i.e: Ctrl+Shift+A-E). It offers access to the most common settings.

To get help on any setting, keep the mouse static for a while after moving the mouse pointer over it . A Tooltip with description text should pop-up. After changing settings, hit Apply and then OK.

#### Tips

- To disable a shortcut , set it to *None* . You can do this by focusing cursor in the Shortcut box and then tapping Backspace once.
- Keep Threshold = 1 to have the Maximum number of active clipboards equal to the Minimum number of Active clipboards.
- To reset the last formatting used in Paste mode (Ctrl+V), check the option "Always start paste mode with the default formatting".
- If you don't like holding down Ctrl in Paste Mode, check the option Start Paste mode with search enabled to start paste mode **with search box**.
- Avoid using Alt only shortcuts as they can cause conflicts with other applications.

## Alternative ways to change settings

If you want to change <u>advanced settings</u>, the Settings Editor is not able to do this and you have to use different tools:

- **Clipjump plugin: IniEditor** As part of the official distribution Clipjump provides a generic editor for INI-Files as a plugin. This plugin named IniEditor can be used to edit settings.ini as well as any other ini-File.
- **Standard Text-Editor** If you want to manipulate some more advanced settings (see below), you may directly open the settings.ini file via a plain text editor

**Translate** 

## Settings.ini

## General

The file settings.ini is used to store the settings of ClipJump. It's located in the same directory where the Clipjump executable resides. Clipjump provides <u>several methods</u> to manipulate the settings file

## **Advanced Settings**

# Setting shortcut to copy to system Clipboard (bypassing Clipjump)

You can also create a unique shortcut in Clipjump to copy/cut data only to System Clipboard and not Clipjump. To create these shortcuts, open the [Advanced] section of Settings.ini and write down shortcut values for *windows\_copy\_shortcut* and *windows\_cut\_shortcut*. As you may have guessed, these shortcuts are disabled by default and by writing shortcut values for them in the ini you create them.

These shortcuts can be very useful when in need to copy sensitive or crap information of the Clipboard.

#### **Changing Action Mode Easy keys**

Action Mode keys can be changed through <u>ClipjumpCustom.ini</u>, please see <u>this</u> example.

To disable a key, empty its value. In the following example, F1 AND L keys are removed.

[actionmode\_delete\_bad\_keys]
ACTIONMODE.F1=
ACTIONMODE.L=

#### Using Win as Paste Mode key instead of Ctrl [v12.3+]

To use Win instead of Ctrl for activating and controlling Paste Mode, set the option *WinForPasteMode* = 1 in the *[Advanced]* section of settings.ini. It is recommended to use the plugin <u>IniEditor</u> to do so. Don't forget to restart Clipjump after updating Settings.ini.

#### Using Win-C and Win-X for copy and cut to Clipjump

#### [v12.5+]

With the setting "Use Win-based shortcut for copy/cut to Clipjump" in the Settings Editor, you can copy/cut to Clipjump using Win-C and Win-X respectively. Now, Ctrl-C and Ctrl-X will have no effect on Clipjump and the clipboard changes through them will be ignored. This setting changes the Using Win as Paste Mode key (Link) setting with it. So you can use the usual Ctrl-C,X,V for interacting with system clipboard and Win-C,X,V for interacting with Clipjump clipboards.

#### Showing the Pasting tooltip

You can make Clipjump show a Pasting.. tip when you paste some data. To do so, change the value of *Show\_pasting\_tip* in the *[Advanced]* section of Settings.ini to 1.

#### **Changing default Task Priority**

You can change default Clipjump Task priority from the Priority key in the System section of the Ini. Set Priority to H (high) to make Clipjump most responsive. Priority Values can be-

L (or Low), B (or BelowNormal), N (or Normal), A (or AboveNormal), H

# Capture clipboard changes only through Ctrl-C and X [v12.5+]

Changing the setting *monitorClipboard* in the *Main* section of Settings.ini to *0* will make Clipjump ignore any clipboard change other than one done by Ctrl-C or Ctrl-X. (or Win-C and Win-X if you have enabled the <u>setting</u>)

#### Make Clipjump flush RAM contents to disk [v12+]

The *RAM\_flush* option when enabled instructs Clipjump to flush its RAM memory data to hard disk at needed intervals. Thus Clipjump will take much less RAM (<= 1.5 MB) but the overall performance may be affected. I recommend keeping this option disabled (set to 0) which is what it is by default. You can find *RAM\_flush* in [System] section of Settings.ini

#### Show paste mode at fixed position

You can fix the Paste Mode's screen position by adding values to variables *pstMode\_X* and *pstMode\_Y* in the Advanced section of the Ini. The X coord is pixels from left of the screen whereas the Y coord is pixels from top of the screen.

You can also leave either of the coords blank and only the specified axes (X or Y) will be fixed for the paste mode.

**Translate** 

## **Translations**

- <u>Overview</u>
- <u>Getting it in your Language</u>
  <u>Translating</u>

## Overview

By default, Clipjump starts up in English and is also available officialy in the following languages.

Language	Translator
Chinese	tuzi
German	fump2000, hoppfrosch
Swedish	Åke Engelbrektson
Russian	maksnogin
Dutch (Nederlands)	

Fred Van Waard Portuguese (Brazilian)Leonardo Lehnemann These Langauge files are located in the languages folder of Clipjump's directory and are merely text files having a ini-type var=value structure. To change languages, double click Clipjump's Tray icon or right-click it and select About Clipjump. There you will be find the list of all the language files found in the languages folder. Select one and it is automatically applied, no restart needed.

## **Getting it in your Language**

Clipjump has also been translated in many other major languages like French, Japanese, Spanish, Italian and Russian by the author using his program <u>Ini Translator</u>. If you want to download any of the above listed translation files, the official <u>Clipjump translations page</u> lists them all and they can be downloaded on-demand.

### Translating

If you look forward to add your language to Clipjump, I will be happy to recieve it and make it available in the future versions of Clipjump. To start off, you can use the <u>Ini Translator</u> program to create a base translation file from the languages/english.txt file in your language to work on. Then you can correct the few grammar mistakes the Translator may have made in the translation and <u>send it to me</u>.

Here is the Preserve keywords list that I use with Ini Translator -

NO-FORMATTING Clipboard Insta-Paste Action Mode Channels Channel Ignore(d) Windows One Time Stop Ignore Windows Manager languages/english.txt pitswap

**Note** - Consider saving the translation file in UTF-8 BOM encoding when translating a Unicode language.

<u>Translate</u>

## Custom

#### [v10.6+]

*ClipjumpCustom.ini* is the medium to create scripts for Clipjump. Usually a script has 3-5 lines of code but can accomplish a lot. The file is created and loaded when you start Clipjump. It gives power to-

- Change previously unchangeable shortcuts
- bind a hotkey to an action (like a hotkey to toggle Incognito Mode)
- run inbuilt *labels* and *functions* to tailor-make any complex user function.
- run a plugin at will or at required condition
- send key-presses to facilitate scripting / macro

## The basics

**Clipjump Custom** works on the principle of Sections ([section\_name]) in Ini's. Each section in ClipjumpCustom.ini behaves as an independent script. You can add commands which is a key=value pair in these sections and they will be executed one by one when the section runs.

The sections are of 2 types - Auto-executing and Non-auto-executing . The only difference between them, as the name suggests is that an *Auto-executing* section runs at Clipjump startup whereas it is not so for the other type.

A section becomes *non-auto-executing* only when it has a hotkey linked to it or when *noautorun* is set to 1 in that section. You will learn more on this topic later.

#### How it works ?

*ClipjumpCustom.ini* is interpreted by Clipjump at its startup. The ini is scanned for auto-executing sections and these are executed as soon as they are found. All sections are then kept in memory and executed later on when called.

So if you edit the ClipjumpCustom.ini while Clipjump is running, the change is not recorded by Clipjump. Therefore you need to reload *ClipjumpCustom.ini* from the tray menu.

Here is an <u>example *ClipjumpCustom.ini* file</u>

## Commands

Here is the list of commands you can use in your ClipjumpCustom section (script). The syntax of all of these commands is *command* = value.

run

Runs a label or function or API function distributed with Clipjump. For a complete list of these, see <u>API functions and routines list</u>. See an example of using Run command <u>here</u>

bind

Binds a hotkey to the current section. This hotkey when pressed will execute that section. The bind command or rather the bind *key* can be present anywhere in the section to be recognizable.

It should also be noted that a section having a bind key doesn't autorun with Clipjump i.e. the section becomes non-auto-executing type. See <u>this</u> example using bind.

tip

Shows a tooltip with the text which is passed as value.

Example-tip = some text to show as tip

send

Artificially presses the key sequence that is passed as value. Example -

send = first line{enter}second line{tab}also in second line

The syntax is based on the AutoHotkey's <u>send command</u>. Use braces {} when you want to press the actual key.

Example - send = {Enter} and send = Enter produce different results. The first one presses the key Enter whereas the second one writes "Enter".

See <u>this</u> example on send to learn more.

sleep

Pauses the execution of section for the passed time in milliseconds. Example - sleep = 1500 pauses the execution of running thread for 1500 milliseconds i.e 1.5 seconds

noautorun

If this variable is set to 1 anywhere in the section, the section **doesn't auto-execute** even without using the <u>bind</u> command.

Example - *noautorun*=1

Set this variable to 0 in a non-auto-executing section ( like the one using *bind* command ) to make it autorun.

Besides this you can also change the inner variables of Clipjump by using the variable = value syntax. Some of the most common variables needed to work highly requested features have been given in the following examples. Go through them and explore interesting tweaks and possibilities. Note - The option to reload *ClipjumpCustom.ini* provided in the tray menu doesn't execute the auto-executing sections of the ini. If you change/add an auto-executing section in the ini, you will have to restart Clipjump to load them.

### Examples

#### Example 1 (Shortcut to toggle Incognito Mode)

```
[toggle_incognito]
bind=Win+Alt+I
run=incognito
tip=Incognito Mode toggled
```

The above code uses 3 commands, bind, run and tip.

- 1. <u>Bind</u> creates a shortcut for the section meaning that the section will not **auto-execute at Clipjump start**. Here that shortcut is Win + Alt + I.
- 2. <u>run</u> runs the label 'incognito' inside Clipjump which is responsible for toggling incognito mode. More labels are <u>here</u>
- 3. <u>tip</u> shows a tooltip when this custom function is run. Here the tip shown will contain the text Incognito Mode toggled.

Note that you can also use noautorun=1 to prevent auto-execution of section even if it doesn't have a 'bind'.

#### Example 2 (changing paste-format key to Y)

[AutoRun] pastemodekey.z=y

The above code changes the inbuilt variable pastemodekey.z to 'y'. If you remember, z is the default key for toggling <u>Paste formats</u> in <u>Paste</u> <u>Mode</u>. The above code changes it to Y. It is mandatory to write pastemodekey in pastemodekey.z because pastemodekey is the object name which has this setting.

The Pastemode group has all variables as pastemodekey.*key* where *key* is the original key. Same is for [Search-Paste Mode]. It has all variables as spmkey.*key*. So the following code will change Channel up, Channel down and paste-format shortcut in [Paste-Mode] and cancel shortcut in [Search-Paste Mode].

```
[AutoRun]
pastemodekey.z=y
pastemodekey.up=q
pastemodekey.down=d
spmkey.home=PgDn
```

The key names are according to the <u>AutoHotkey key list</u>.

#### **Example 3 (Shortcut to check for updates)**

```
[section_name_can_be_anything]
run=updt
bind=Win+Alt+U
```

- *updt* is label used in Clipjump to update and the <u>run</u> command runs it.
- As you see, it is not necessary to keep <u>bind</u> at the first line of the section. It can be anywhere.

# Example 4 (Shortcut to paste current content on system clipboard instantly [without formatting])

```
[paste_current]
bind=Ctrl+Shift+V
run=simplePaste
[paste_without_formatting]
bind=Win+V
run=API.runPlugin("noformatting_paste.ahk")
```

The above code has 2 examples, [paste\_current] is to paste current clipboard as it is and [paste\_without\_formatting] is to paste current clipboard without formatting.

- *simplePaste* is label in Clipjump to paste current content on Clipboard.
- API.runPlugin( plugin\_path ) runs a plugin file. Here in the 2nd section, it runs the plugin file *noformatting\_paste.ahk* which is supposed to contain the code to paste without formatting. Learn more

about <u>API.runPlugin()</u>.

• Another thing to note here is the use of quotes around noformatting\_paste.ahk. As in a function call, arguments are separated by comma (, ), so it is advisable to use quotes around arguments to separate literal comma from the one meant to partition arguments.

#### **Example 5 (Instantly switching to your favorite channel)**

Suppose your favorite channel is #4 as it stores some important web links. We will use the inbuilt changeChannel function in Clipjump to do the purpose.

```
[changeto_4]
bind=Win+4
run=changeChannel(4)
tip=Channel changed to 4
```

# Example 6 (Running a ClipjumpCustom script with Action Mode)

Suppose you don't want to create a separate shortcut Win+V for the above example. Here is how to create a [Action Mode] shortcut for it.

```
[changeto_4]
noautorun=1
; noautorun prevents auto-execution of a section in absence of t
run=changeChannel(4)
tip=Channel changed to 4
[actionmode_f10]
; here I don't use noautorun as I want it to run at startup and
ACTIONMODE.F10 = API.executeSection(changeto_4)
ACTIONMODE.F10_caption = Change 2 my channel
```

- API.executeSection( section\_name ) can be used to execute any section any time. section\_name is the name of the section to be executed.
- [Action-Mode] keys are customizable with the ACTIONMODE.key variable. The ACTIONMODE.key\_caption variable holds the caption for the action shown in [Action-Mode].
- The ACTIONMODE.*key* on the other hand holds the label/function/routine to be executed when the key is pressed. Here we are executing the function API.executeSection() with the first parameter as *changeto\_4*

#### **Example 7 (Paste Your Contact Details)**

If you give a unique tag to a clip with some specific information, then it can be retrieved anytime using the <u>getclipDataByTag()</u> function and pasted using the <u>pasteText()</u> function.

```
[paste_address]
bind=Ctrl+F8
zAddr=%API.getClipDataByTag(my_address)%
run=API.pasteText(%zAddr%)
```

• Here *my\_address* is the tag that your contact details clip has.

Note It must be noted that Clipjump doesn't support nesting of %..%. So something like zAddr = %API.getClipDataByTag(%some\_variable%)% is invalid. However the output of <u>run</u> command is stored in variable ans.So the above code can be written as -

```
[paste_address]
bind=Ctrl+F8
run=API.getClipDataByTag(my_address)
; now you can use API.getClipDataByTag(%some_variable%)
run=API.pasteText(%ans%)
```

#### Example 7.5 (Getting the clip to paste)

You can use the inbuilt inputbox() function to retrieve the *clipno* to paste. This procedure can also be done for the *channelno*.

```
[paste_sig]
bind = Ctrl+F8
zClipno = 1
; zClipno = %inputbox("Choose", "Please write the clip number tc
run = API.paste( 2, %zClipno% )
```

- In the line zClipno = 1, zClipno is a user created variable that is assigned the value of 1.
- The variable is fed into API.paste() function by wrapping it with %....% . So you can say that %...% is used to declare that something is a variable i.e. NOT Literal TEXT.
- In the commented line containing %inputbox...% the variable zClipno is assigned the return value of inputbox() function. As the whole function call is NOT supposed to be A FIXED TEXT, we have used %...% over it. You can try un-commenting that line and then restarting Clipjump to see what happens.
- inputbox( title , caption\_text ) is basically a function that shows a dialog box for the user to enter text. Once text is entered and OK is pressed, the function returns the text entered by user to the variable calling it.

#### Example 8 [SEND] (Filling Form)

The first example using <u>Send</u>. As briefed above, the command sends or stimulates a key press.

So to get started, know that 'Send' has some rules which must be noted.

- To simulate key combination press like Win + C, use send=Win+C. When pressing key combination using send, keep it in a single line.
- To simulate a key press like Tab or PageDown, use send={Tab} or send= {Enter} or send={tab}{enter} if you want both.
- To write text using send command like 'Hi there', use send=Hi there.

**Situation** - Suppose you want to fill a web form with 4 fields and the fields focus can be switched with TAB, here is the customization to use -

```
[fill_that_form]
bind=Ctrl+Shift+F9
run=API.paste(,1)
send={tab}
run=API.paste(,2)
send={tab}
run=API.paste(,3)
send={tab}
run=API.paste(,4)
```

- Keep focus on the first of the 4 fields and press Ctrl + Shift + F9 and watch all the other 4 fields filled automatically.
- As you may have read in example 7, API.paste() pastes clipno from channelno. If channelno is blank (like here), it defaults to the current active channel.
- You may use sleep=500 to wait for a particular time in milliseconds.

#### Example 9 (moving clips)

We will use the API.manageClip(new\_channel, channel, clip, flag) function here. *channel* and *clip* default to the current active ones if no value for them is passed.

**Situation** - Suppose you copied a trash / incorrect information in the current protected / normal channel accidentally. Here is the code to move it to <u>Pit</u> channel which is say at channel number 2

```
[trash_clip]
bind=Ctrl+Win+Del
run=API.manageClip(2,,,0)
tip=The current active clip was trashed
```

The last parameter *flag* is 0 for move and 1 for copy. Here we are moving the clip.

# Example 10 (paste mode with search enabled; no need to hold Ctrl)

If you don't like holding down Ctrl to navigate clips, this if for you. This Customization will bind Win + V to open paste mode with Search enabled so you will be able to use Up/Down keys to navigate through the clips.

```
[paste_with_search]
bind=Win+V
spm.active=1
run=paste
run=searchpm
```

#### Example 11 (emptying a channel)

```
[empty_pit]
bind=Win+Alt+E
run=API.emptyChannel(pit)
tip=Pit Channel is now empty
```

The above uses the API.emptyChannel(channel\_no) function. If *channel\_no* is empty i.e nothing is passed as this parameter , the current active channel is taken into account. If *channel\_no* is not number, a channel with name as *channel\_no* is taken into account. If *channel\_no* is number, it is supposed that *channel\_no* contains a channel number to be used.

#### **Example 12 (Variables concept and STORE.)**

```
[SEC1]
STORE.myVar = text in store.myvar
anotherVar = text in anotherVar
tip = %STORE.myVar%
sleep = 1000
tip = %anotherVar%
```

```
sleep = 1000
[SEC2]
tip = Sec2 is now started
sleep = 1000
run = API.showTip(%STORE.myVar%, 1000)
tip = %anotherVar%
sleep = 1000
```

- STORE.*varname* can be used by users to create variables that are saved all through Clipjump runtime and can be accessed anywhere. In the above code, both SEC1 AND SEC2 neither have bind nor noautorun=1 so they will run at Clipjump startup.
- SEC1 will run before SEC2 and will add values to the variables 'STORE.myVar' and 'anotherVar'.
- A tip confirms what the value is stored in these variables. Note that tip default shows for 1000 milliseconds (1 second) and so we use sleep=1000 to pause execution for that time so that tip is properly
  shown. You can also use the function API.showTip(text,
  time\_in\_ms) as I have used in SEC2 as
  'API.showTip(%STORE.myVar%, 1000)'.
- When the above customization runs, you will see that tip containing *%anotherVar%* is not shown in SEC2. It is because *%anotherVar%* is unaffective there. It's lifetime was only till SEC1 where it was created.
- %STORE.myVar% on the other hand will be active at all times and you will see its tip in SEC2.

Note - A list of all API functions, key label, routines and general functions and Interesing Variables can be found on <u>Developer's Reference Page</u>. It is recommended to have a look at them when you complete this guide.

If you need help on Customizations, you can ask in comments on the <u>main</u> <u>site</u> or from any of links in the <u>contact</u> page.

**Translate** 

# **HotPaste Plugin**

- IntroductionBasic syntax and Usage
- Examples
  End notes

# Introduction

HotPaste plugin, an idea inspired from AutoHotkey hotstrings helps you create hot-words that perform action when you type. The basic actions can be pasting a text from a Clipjump channel or through a function.

Raw strings can be pasted from API.pasteText() function whereas clips can be pasted from API.paste(channel, clip) function. The pasteText() will only be able to paste unformatted text so for any HTML or picture type of clip, you will have to rely on paste().

You can create Hotpastes (i.e. "Hot-words") in the

*plugins\hotpaste.lib\base.ahk* file. This file in **not distributed** by default and you will have to run **HotPaste** from the Plugin Manager to have this file created. Once created, you will have to edit the file to add your HotPastes. After editing *base.ahk*, **restart Clipjump** to have your edits loaded.

#### **Basic syntax and Usage**

A simple *base.ahk* goes as-

The above code creates two HotPastes - cj\_site and my\_add.

- As the comments read, writing cj\_site and then pressing Enter or Space or Tab will replace the typed 'cj\_site' by http://clipjump.sourceforge.net. The <u>PasteText()</u> function is responsible for pasting this plain text. Note that I have surrounded the literal string in quotes ("").
- The 2nd HotPaste my\_add when typed invokes <a href="mailto:Paste(">Paste()</a> function of the API to paste clip 2 of channel 0. The typed text 'my\_add' will be replaced by the clip 2 of channel 0 in this case. As this time there is no literal string as input, I have not used quotes.

So the basic syntax is -

```
::HOTPASTE::
API.Paste or API.pasteText
return
```

If you have knoweledge of <u>AutoHotkey</u>, there is no limit to what HotPaste can do for you. After all, HotPaste is a simple plugin that uses the core feature of AutoHotkey for its purpose.

BTW, this doesn't mean you will have to learn AutoHotkey (AHK) to use it. This simple tutorial will be sufficient.

#### Examples

#### **Pasting multi-line text**

Multiline separator in Autohotkey is `n.

#### Pasting 2nd clip of current channel

CN.NG variable contains the current active channel number. See <u>more</u> <u>variables</u>.

```
::2ndlast::
API.Paste( CN.NG, 2 )
return
```

#### **End notes**

- Make sure you **resart Clipjump** after changing *base.ahk* to have the changes loaded.
- It will be a good idea to <u>fix</u> clips that you access through API.Paste() because of the simple reason that clip number will change on addition of new clips. It would be better if you maintain all such FIXED clips in an entirely different channel to avoid workflow problems.

**Translate** 

# **Plugin Development**

- 1. Introduction
- 2. <u>The filename</u>
- 3. <u>Plugin parameters</u>
- 4. <u>Script/Function parameters</u>
- 5. <u>Conventions</u>
- 6. Plugin Classes
  - pformat
  - <u>external</u>
- 7. Things to Note
- 8. <u>Libs</u>
- 9. Developers Complete List

# Introduction

Plugins are AutoHotkey scripts that are run by the Clipjump.exe which is nothing but AutoHotkey.exe in disguise.

Some plugins are included in the program at runtime, some are not (external class). There is no limit on the number of plugins and they can be always accessed from the Plugin Manager (ActionMode = M)

## The Filename

The filename of the plugin symbolizes the function it is supposed to contain. If filename is *myplugin.ahk*, it means the plugin file contains a plugin\_myplugin() that is run by the Plugin Manager or API.runPlugin(). If the filename is *pformat.noformatting.ahk*, the main function the file has is plugin\_pformat\_noformatting().

# **Plugin Parameters**

If you open any of the officially distributed plugins in Notepad, you will see lines starting with ;@Plugin-. These hold the key-value pairs that have information about the Plugins.

So ;@Plugin-Name myplugin gives the key *name* for the plugin the value *myplugin*. As you may have guessed this is the same thing you see when you click on Properties in the Plugin Manager.

There is no limit on what can be the 'key' here. Even ;@Plugin-authorSite http://mysite.com also works and is shown in the Properties window.

# **Script/Function parameters**

The Plugin parameters like ;@Plugin-param(N) where *N* is a natural number are the Function parameters (like ;@Plugin-Param1). They hold information about the *Nth* parameter of the main function in the plugin file. When you run a plugin with API.runPlugin( plugin\_filename , param1, param2 ....) without any 'param's , the function auto-asks you the parameters showing these informations if they are present. You can provide a parameter as ;@Plugin-Silent 1 to force API.runPlugin() to **not ask** the user for parameters even if no parameter is passed to it.

Function parameters are not Mandatory but it is recommended to provide them.

## Conventions

- 1. Variable Names in plugin files that are included (not external) should start with *z*. This is because Clipjump doesn't uses any variable starting with 'z' and so this is the best option to avoid variable conflicts.
- The parameter names in the function definition ( plugin\_myplugin(zParam1, zParam2)) should also start with 'z'.

**TIP-** You can use the STORE object to also store variables. It is a **global** object and is totally meant for storing by plugins and through <u>ClipjumpCustom.ini</u>. Eg> STORE.myvar := "text" and zVar := STORE.myvar

# **Plugin Classes**

'Classes' is a way to distinguish certain plugins from normal plugins. All the plugins whose name is like *a.b.ahk* and not *b.ahk* mean they belong to a class (here *a*).

As already said, these files have main function like plugin\_a\_b(). Currently Clipjump has 2 defined classes. User is not allowed to create them. (i.e. create a file like xyzPlugin.coreScript.ahk)

#### pformat

The plugins in this class are used in the paste-format option in Paste mode (key = Z). Officially distributed plugins like *pformat.sentencecase.ahk* and *pformat.noformatting.ahk* belong to this class.

- 1. These plugins also have a custom mandatory parameter ;@Plugin-Previewable which should hold 1 if the paste format is previewable in Paste Mode. It is 1 for SentenceCase.
- 2. These Plugins should make STORE.ClipboardChanged true if they have successfully changed the input variable and thus what is going to be pasted.

For example STORE.ClipboardChanged := true

- 3. It should be noted that these plugins (should) have 1 input parameter. See the file 'pformat.sentencecase.ahk' for example.
- 4. When toggling and when pasting, Clipjump inputs the *Clipboard* variable and not the *ClipboardAll* variable as the first parameter. If the plugin needs the *ClipboardAll* variable, it can consult it anytime as it will be the same as what current active

#### external

Plugins whose name starts with *external*. belong to the 'external' class. They are not included in Clipjump at runtime and solely run through the AutoHotkey.exe (i.e. Clipjump.exe) when needed.

The Parameters in external plugins ? Like Function parameters , the

external script file can contain -Param1, -Param2 and so.. for the API.runPlugin() to take them as Script parameters and thus ask the user if they pass no paramter.

For example try running the *external.controller.ahk* and then seeing its source code.

#### Things to Note

- 1. Use ;@Plugin-Silent 1 to make API.runPlugin() not ask the user for parameters even if user doesn't enters them.
- 2. It will be a good idea to use API.showTip() and API.removeTip()
   (Link)
- 3. While testing keep in mind that your plugin will be included in Clipjump.ahk and thus run from the directory of Clipjump.ahk. So be careful with the WorkingDir factor.
- 4. There will be no auto-execution of any plugin as a non-external plugin will be included at the bottom of Clipjump.ahk
- 5. Please reload Clipjump when you change/add a plugin file to load it into Clipjump.
- 6. Make sure to have a look at <u>The Complete Developer List</u> and the <u>public API</u>

# Libs

If your plugin file requires more files as #Includes, consider keeping them in (file\_name).lib directory. So for the plugin *external.controller.ahk* we have *external.controller.lib* as the lib directory. This makes the Delete Plugin option in Plugin Manager work.

<u>Translate</u>

# **Developer's API**

- <u>API functions</u>
- More functions
- Labels/routines
- <u>Variables</u>
- The Public API

## **API functions**

They should be accessed as API.function\_name(). For example API.paste()

- paste(channel="", clipno="")

   Pastes clips from a certain postion in a certain channel. If both are blank, paste 1st clip from currently active channel. Example API.paste(1, 14) pastes clip 14 of channel 1
- 2. getClipAt(channel=0, clipno=1, toreturn=1)
   Get a Clip's content i.e. what is contained in that clip. 'channel' and
   'clipno' are the channel number and clip number respectively.
   If toreturn = 1 then return Clipboard text data
   If toreturn = 2 then return ClipboardAll binary data.
- 3. manageClip(new\_channel=0, channel="", clip="", flag=0)
   Cut/Copy Clip from one channel to other. 'new\_channel' is the new channel to place the clip in. Note that the clip will always be placed at clip 1 of new channel.
  If flag = 0 then cut, if flag = 1 then copy
  If channel is empty, active channel is used.
  If clip is empty, active clip in [Paste-Mode] (Clip x of y, "x") is used.
- emptyChannel(chno="")

- Empties a particular channel (chno)

If 'chno' is empty, it DEFAULTS to current channel.

You can also use name of a channel like API.emptyChannel(web) or API.emptyChannel(pit) or API.emptyChannel(default). Note that if you are creating an AHK Script i.e. a plugin you will have to surround the channel name by quotes ("") to take it as string.

- 5. runPlugin(filename, parameters\*)- Runs a plugin. See here and then here.
- runFunction(function\_string)
  - Runs a <u>function</u>

function\_string - The string containing the function with parameters, like changeChannel(1), inputBox(mytitle, some text to desc) Example - API.runFunction("changeChannel(1)") Note - <u>ClipjumpCustom</u> users are requested to use run=changeChannel(1) rather than this.

disable(status)

Disables/Enables/Toggles Clipjump
If status=1, then disable clipjump.
If status=0, then enable clipjump.
If status=-1, then toggle clipjump status.

- 8. addClip(channel, input)
   Adds a clip at the start of the channel i.e. as Clip 1 channel The *channel* where the new clip is to be added input The text that the new clip will contain
- showPasteTipAt(channel, clip)

   Opens [Paste-Mode] with clip number 'clip' of channel 'channel' active.
- 10. ExecuteSection(secName)
   Executes a section in the <u>ClipjumpCustom.ini</u>
- 11. IncognitoMode(p=1)
   Enable/disable Incognito Mode. Pass 1 to disable.
   If p = 1 enable incognito mode
   If p = 0 disable
- 12. getClipLoc(channel="", clipno="")
   get a Clip exact file location or the file in which the clip is saved channel The channel the clip belongs to clipno Clip number of the Clip
- 13. getClipDataByTag(tag)- returns the text data of first clip which has the tag *tag*
- 14. blockMonitoring(yes=1)blocks Clipjump Clipboard monitoring. Useful when you are changing

clipboard content such that you don't want it to be accounted by Clipjump If first parameter i.e. yes = 1 , then block Monitoring. If 0 , then unblock (or enable Clipboard monitoring)

15. runLabel(name)

- runs a <u>label/routine</u> name - The name of the label to be run.

- 16. deleteClip(channel, clip)
   deletes a clip from a channel
  channel The channel number whose clip is to be deleted
  clip The Clip number to be deleted.
- 17. showTip(Text, forTime="")

- shows a Tip (with a different font). recommended when creating a plugin

Text - the text to show in the tip

forTime - time for which the tip is to be shown, if the paramter is not passed i.e. blank, the tip becomes permanent

18. removeTip()

- removes the Plugin API Tip i.e. the tip created using the above function API.showTip()

19. pasteText(Text)

- pastes the 'Text' that is passed to the function. Text - the text/string you want to paste

- 20. getChStrength(channel)gets total number of clips in a channel i.e. population of a channel
- 21. Text2Binary(Text, byref returnVar)

- gets binary ClipboardAll data from simple text i.e. string. You pass it a string and it converts it into binary data and returns it via byref variable returnVar

Text = the string you want to convert to ClipboardAll type data returnVar = byref variable to return ClipboardAll data

- 22. getVar(var) and setVar(var, value)
  - gets the value of variable 'var'
  - sets the value of variable 'var' to 'value'

### **More functions**

Other interesting functions that are not a part of API. Unlike API, they should be accessed normally as function\_name(). For example changeChannel(2)

- 1. BeepAt(value, freq, duration=150)
   Make a beep system sound.
   If 'value' = 1 , then the function will beep else it will not.
   freq frequency of the beep
   duration duration of the beep to be played in milliseconds
- 2. getClipboardFormat(type=1)

gets the type of data on Clipboard
If 'type' = 1 , then the function returns the type of data a/c <u>Paste-Mode</u>
i.e. "[Text]" for text data including HTML , "[File/Folder]" for
file,folder copied data and (blank) for Picture type data.
If 'type' = 0 , then the function returns the data type formats ID's on
Clipboard as per the Microsoft list.

- 3. inputBox(title, text)
   shows a dialog for user to enter any text and the function returns that
   text. See More
- HParse(hotkey)

- converts a normal hotkey like 'Win+Alt+Z' or 'Ctrl+PrintScreen' to AutoHotkey format like '#!z' or '^Printscreen'. hotkey - The Standard hotkey to convert to AHK form. The function returns the AHK formatted hotkey of the input hotkey.

- changeChannel(channelno)
   Changes channel to 'channelno'
- 6. channel\_find( name )- Finds the channel corresponding to the 'name' and returns its channel number.

name - The name of the channel like 'Default'

7. chooseChannelgui()

- Shows a GUI for the user to select channel, if he selects a channel the function returns the selected channel number.

USAGE-ClipjumpCustom tip = %chooseChannelgui()% USAGE-AhkScript(Plugin) zSelectedCh := chooseChannelgui()

8. manageChannel(channel)

- Deletes a channel with the channel number 'channel' channel - The Channel Number of the channel to be deleted.

9. set\_pformat( paste\_format )

- Sets the Paste format which is currently active. paste\_format - The name of paste format (like 'UPPERCASE') to be made active.

USAGE - set\_pformat(SentenceCase)

#### Labels/Routines

Labels should be accessed as run = label\_name in <u>ClipjumpCustom</u> and gosub label\_name in AhkScript.

- 1. These are the labels to open various dialogs and gui in Clipjump. You can easily guess their functions from their names.
  - 1. actionmode
  - 2. settings
  - 3. history
  - 4. channelgui
  - 5. classtool
  - 6. channelGUI
  - 7. pluginManagerGui
  - 8. channelOrganizer
- 2. These are the labels corresponding to various features of Clipjump. Again, the name itself should be explanatory.
  - 1. onetime
  - 2. holdClip
  - 3. pitswap
  - 4. incognito
- 3. disable\_clipjump

- Toggles Clipjump Status i.e. Enabled or disabled (same like Options > Disable Clipjump in Tray Menu)

- 4. endPasteMode
  - Ends [Paste-Mode] forcibly.
- 5. *simplePaste* 
  - Pastes the current active content on System Clipboard
- 6. windows\_copy and windows\_cut

- Copies / Cuts the selected text or file without copying to Clipjump i.e. only System Clipboard is affected. This is what is used inside <u>Copy to</u> <u>System Clipboard bypassing Clipjump</u>

7. *updt*- Check for Updates

#### 8. *exit*

- Exits Clipjump properly

# Variables

The important variables in Clipjump which can be changed to customize much of the behavior in the program are listed below. Please note that only selected variables have been listed to prevent any issues with the core functionality of Clipjump.

Changing a variable in <u>ClipjumpCustom</u> is done as variable = value and in AHKScript is done as variable := "value"

- 1. Channel Object
  - 1. *CN.NG* Holds the channel number that is currently active.
  - 2. *CN.Name* The name of the channel that is currently active.
  - 3. *CN.Total* Total number of channels that are currently created by the user.
- 2. STORE Object

- Public Storage Space. Use as STORE.var := "value" and msgbox % STORE.var

See also **<u>STORE in ClipjumpCustom</u>**.

- 3. *curPformat* Holds the current active paste format.
- 4. *CALLER* Holds 1 if clipboard is monitored by Clipjump and 0 if not.
- 5. *pastemodekey* and *spmkey*

- Use as pastemodekey.*key* OR spmkey.*key* to change keys currently used in [Paste Mode] and [Search-Paste Mode]. See <u>here</u> for an example.

### **Public API**

Public API (publicAPI.ahk) gives power to any Ahk Script as well an <u>external plugin</u> to control all aspects of Clipjump from changing and reading variables to running function and labels. It is by default distibuted with Clipjump and present in the Clipjump.exe directory. <u>Here</u> is the online github link in case it ever updates

An external plugin can include publicapi as #Include %A\_ScriptDir%\..\publicAPI.ahk Once publicAPI is included in the script, you will have to initialize it.

```
cj := new Clipjump() ; The name 'cj' doesnt matter
```

Then you can call API methods, set variables, get variables and so on using the 'cj' object. Here are few examples -

```
; getting value of internal variable VERSION
msgbox % cj.version
; getting value of internal variable CN.NG
msgbox % cj["CN.NG"]
; setting internal variable PASTEMODEKEY.Z to Y
cj["pastemodekey.z"] := "y"
; running a <u>API</u> function GETCLIPLOC
msgbox % cj.getClipLoc(1, 5)
; running a normal function SET_PFORMAT that is not part of API
cj.runFunction("set_pformat(NO-FORMATTING)")
; running the label EXIT
cj.runLabel("exit") ; closes Clipjump
```

The plugin Ignored Windows Manager uses the publicAPI. It will be a good idea to read its code (*external.ignoreWManager.ahk*).

#### **Translate**

# **Shortcuts List**

- <u>General</u>
- <u>Paste Mode</u>
  - <u>Search-Paste Mode</u>
  - Cancel, Delete, Copy, Move and Delete All Mode
- <u>Action Mode</u>

# **General (System Wide)**

Ctrl + VPaste ModeCtrl + Shift + AAction Mode

#### **Paste Mode**

Ctrl + v is used to enter the **[Paste Mode]** (see also <u>here</u>). As releasing both keys inserts the current displayed clip, the key sequence

- Press Ctrl + V
- Keep Ctrl pressed
- Release v

enters the intrinsic [Paste Mode]. While being in Paste Mode (Ctrl pressed) a few more actions can be performed by pressing keys:

```
V - Move forward through Multiple Clipboards (from 1 to 2 , 2 to 3,
C - Move backward through Multiple Clipboards (from 3 to 2 , 2 to 1,
S - Add current Clip to Windows Clipboard
X - Toggle between Cancel, Delete, Copy, Move and Delete All MODE
E - Export current clip
A - Jump to first clip
F - Activate <a>Search / Entering [Search-Paste Mode]</a>
H - Edit Clip
Z - Toggle pasting formats
T - Add <u>tags</u> to clips
Q - Move clip to first position
← (i.e. Enter) - Start <u>Multiple Pasting</u> Session
Space - Fix/un-fix a clipboard at its position (Fixate)
t (i.e. Up) - Change channel to 1 up (+1)
\downarrow (i.e. Down) - Change channel to 1 down (-1)
1..9 - Move to clip {num} ahead/behind of current clip
Shift - Hold Shift with Ctrl to delete clip after pasting. See Paste
F1 - View listing of all keys in Paste Mode
```

See <u>this ClipjumpCustom.ini</u> example to change these keys

#### Search in Paste Mode (Search-Paste Mode)

The following key sequence enters **[Search-Paste Mode]** while being in **[Paste Mode]** (see also here):

- While Ctrl pressed ... (i.e. while being in [Paste Mode])
- Press F

• Release Ctrl and F

Now you are in [Search-Paste Mode] and several commands are available via keypress:

```
Enter - Paste
Home / Esc - Cancel
↑ (i.e. Up) - Move one clip up (+1)
↓ (i.e. Down) - Move one clip down (-1)
Ctrl+[Paste Mode] key - execute a [Paste Mode] feature when [Search-
Ctrl + Enter - [Multipaste Mode] within [Search-Paste Mode]- Paste w
Ctrl + F, then release F - Switch back to [Paste Mode]
```

See <u>ClipjumpCustom.ini</u> to change these keys

#### Cancel, Delete, Copy, Move and Delete All Mode

While being in <u>Paste Mode</u>, pressing x repeatedly toggles between several modes:

```
X - [Cancel Mode] - cancel the current paste operation
X - X - [Delete Mode] - delete current clip
X - X - X - [Move Mode] - Move clip to a different channel
X - X - X - X - [Copy Mode] - copy clip to a different channel
X - X - X - X - X - [Delete All Mode] - deletes all clips
```

Note that [Cancel Mode], [Delete Mode], [Move Mode], [Copy Mode] and [Delete All Mode] are inter-related. The program cycles through these modes when you press X while holding Ctrl.

So, pressing x while holding Ctrl in [Delete All Mode] will switch back to [Cancel Mode].

#### **Action Mode**

Ctrl + Shift + A is used to enter the **[Action Mode]** (see also <u>here</u>). A dialog box pops up, offering a lot more "actions" which can be performed. These actions can be accessed via keypress:

- B Hold Clip C - Copy File path(s) D - <u>Disable Clipjump</u> E - <u>Settings</u> F - <u>Copy File Data</u> F1 - Help F2 - Paste Mode Shortcuts H - <u>Clipboard History</u> L - Ignore Windows Manager M - Plugin Manager 0 - <u>Channel Organizer</u> P - PitSwap T - One Time Stop U - Sync Clipjump Clipboard X - Copy active folder path Esc - Exit Action Mode Window
- 0..9 Activate channel whose ID (number) is pressed

The hotkey for [Action Mode] can be user defined. (see: <u>ClipjumpCustom</u> and <u>Settings Editor</u> to change this key)

To change Action Mode easy keys, see Changing Action mode keys

<u>Translate</u>

# Changelog

# 12.5

#### 14/04/2016

- Fixed crashing issues in v12.3 by using TooltipEx
- Now using SQlite database for History
- Ignore quick (<200ms) clipboard changes to prevent crashes
- [New] Option to use Win-C and Win-X for copying to Clipjump
- [New] Option to disable automatic clipboard monitoring

## 12.3

#### 09/11/2015

- Fixed bug with Action Mode not showing on some systems
- [New] Move clip to first position in Paste Mode using Q
- [New] <u>Pop</u> from channel
- [New] <u>Use Win</u> instead of Ctrl for Paste Mode
- [API] New function : <u>disable</u>

#### 21/05/2015

- Added feature to delete multiple items in Channel Organizer
- Improvement in Common Formats
- Removed Select Channel GUI, Channel Organizer will take its place
- Added Portuguese-Brazilian language
- Many improvements for ClipjumpCustom
- Better keyboard handling for Channel Organizer
- Added new API function getClipByTag()
- Lots and lots of bug fixes

### 12

#### 26/08/2014

- Fixed a bug with hotpaste plugin adding content to clipjump when pasting
- Added new common format UnHTML (thanks to derRaphael for the lib)
- Removed default shortcut win+c for history tool

#### 06/08/2014

- Added new formats, *TrimWhiteSpace* and *DeHTML* in <u>Common</u> <u>Formats</u>
- Added <u>HoldClip</u> in History Tool
- Added option to change channel in <u>channel organizer</u>
- Added option in <u>Protected channel</u> popup to transfer to default channel
- Added option to open clip in Common Format plugin via HoldClip
- History tool can now be kept active after Insta Pasting
- Fixed bug where some icons were not displayed in channel organizer
- More minor bug fixes, doc and program improvements

#### 04/07/2014

- Fixed a bug with Common Formats in the previous versions.
  Updated <u>ClipjumpCustom</u> documentation
  More minor improvements

#### 29/06/2014

- Added tips over Channel organizer buttons
- Added option to always start paste mode with default formatting
- Minor enhancements in documentation, common formats, plugin manager and channel organizer

#### 10/05/2014

- Added create new clip/create new channel feature in Channel Organizer
- Added Edit Clip feature in Channel organizer
- Overall enhancement in channel organizer in all the features

### 11.2.3

#### 06/05/2014

- Use <u>0..9 keys</u> to change channel through Action mode
- Added open in Paste mode option in Channel Organizer
- Fixed few bugs introduced in the previous version

#### 05/05/2014

- You can now add <u>Tags</u> to clips
- Use <u>F1 in paste mode</u> to see list of all feature available
- Use <u>1..9</u> number keys to jump ahead in Paste mode.
- Plugin Manager now returns the output of all plugins through a gui
- Added option to open paste mode with search box enabled
- Enhancement: Preview now wraps clip text
- BugFix: Channel Default was not rename-able
- More minor bug fixes and enhancements

#### 23/04/2014

- Added 'Channel Organizer' to help manage channels through a Graphical interface
- Editing image now supported in Paste Mode
- Fixed Clips not shown correctly in presence of **Clipboard Monitoring program** like Executor.
- Fixed Image not shown in Search in Paste Mode
- Many other small fixes

#### 16/04/2014

- Added Hold Clip feature to copy clip to buffer without copying it to Clipjump
- Added 'RegExReplace' in Common Formats v0.3
- Fixed Image not moving in Copy/Move Clips
- Fixed Fixate not mainatained when clips were copied/moved.
- Re-Arranged the Help File, hopefully more easy to follow now.
- Moved Ignore Windows Manager to external plugin

#### 11/04/2014

- Added "Common Formats" paste format plugin
- Enhanced the PublicAPI and it is now distributed with Clipjump
- Fixed: Issue with Shift+4 shortcut due to Clipjump Custom
- Fixed: Errors due to missing plugins in plugin system

### 10.7.8

#### 27/03/2014

- Fixed paste-formats misbehaving with clip1 when pasting in different formats repeatedly
- Fixed "clips not previewing" issue in presence of high-memory programs like Netbeans
- Improvements in the hotPaste plugin
- Added Image editing support in History Tool
- Other Help File and Action Mode enhancements

### 10.7.5.1

#### 24/03/2014

- Fixed a bug in ClipjumpCustom that arised in the previous version.
- Fixed bug of not deleting when File/Folder clips were continuous by "Delete File/Folder" plugin.

## **10.7.5**

#### 16/3/2014

- Clipjump now supports Plugins
- Custom Paste Formats like NO-FORMATTING are now possible with the added Plugin Support
- Multiple Insta-paste'ing possible in History Tool.
- Use Enter in Paste mode to paste without closing Paste mode. After that, release Ctrl to cancel.
- Use Ctrl+Enter in Search in Paste mode to paste without closing the window i.e. multi-paste.
- ActionMode keys are totally customizable now. Add, remove, modify..
- Added Edit option in History
- Use %...% in Customizer to declare variables.
- Also added function support in %..% variables.
- Use noautorun=1 in a section in ClipjumpCustom.ini to avoid autoexecuting it even if it doesn't have a bind key.
- Added API.executeSection() to execute a section at will in ClipjumpCustom.ini
- Added Store. object (multi-variable holder) to store variables that are valid throughout the Program's runtime.
- Fixes in EditClip option in Paste mode
- **CHANGED** No-Formatting option is now a part of paste-formatting option

- Fixed bug with more than 9 lines in a section in ClipjumpCustom.ini
- Added: a new function API.emptyChannel() for ClipjumpCustomizer
- Tried to fix bugs with Emacs-org-mode
- Minor: little more enhancements in ClipjumpCustomizer

- Fixed issue with Action mode in Russian systemsMinor: fixed about gui title centering

- Minor fix: bug when number of days in history was set to 0
- Minor fix: a clip was FIXED at pos 1
- Major fix: invalid hotkey error when used in Russian systems
- Added: A sound beep in protected channels

- Fixed a bug with A key in Paste mode
- Fixed bugs with channel delete feature
- Added feature to <u>customize + scripting (Clipjump Customizer</u>) in Clipjump
- Added <u>search feature</u> in paste mode
- Added feature to <u>edit clips</u> in Paste mode
- Added feature : <u>Protected Channels</u>
- Clipjump now shows changes in current version when updated

#### 2/1/2014

- Added option to Copy and Move clips from one channel to another
- Now press A in paste mode to go to Clip 1
- Current Clip pointer in paste mode is never discarded
- You can use Middle Mouse Button to Insta-paste
- Added option in Ini to set program's default priority

### **10**

### 9.9.3

#### 17/12/2013

- Fixed bad gui control sizes in all windowsAdded option to delete a channelOther small bug fixes

### 9.9.1

#### 6/12/2013

- Made Clipjump possible to be translated in other languages.
- Removed system-level shortcuts for copy file data, copy file path, pitswap and channel selector. They can be accessed from **Clipjump** Action Mode and a system-level shortcut for them can still be created.
- fixed many odd bugs

- Added Ignore Windows Manager , a GUI for managing Ignored (disabled) windows for Clipjump. (credits to fump2000)
- Partially fixed bugs with Notepad++ column mode and Excel multiplerow/column paste.

- Fixed bug with Clipjump copying in zipped files opened in Windows Explorer
- Fixed some bugs with Excel **2010** and Clipjump

- Fixed bug with Clipjump in MS Excel
- Fixed bug with Cut functionality in MS Excel and Kingsoft Office (you can add more spreadsheet application too)

- Added an option to auto-disable Clipjump when certain window is active
- Added an option to play sound when data copied to Clipjump
- Fixed bugs with Channel cycles in Paste-Mode

### 9.7.2

- Bug fix in No-Formatting option
  Bug fix in the Update Dialog Box
  Added Translate option to the Help File

- Added Action Mode (Ctrl+Shift+A) . One place to execute all the functionalities in Clipjump.
- New 'Disable Clipjump' option available in the tray as well as Action Mode.
- Made Clipjump suitable to work with all keyboard layouts.
- Re-organized Tray menu.
- 'One Time Stop' and 'Copy active folder path' have no default shortcuts now. They can be used from Action Mode as well as their custom shortcut can be created.
- Delete All mode now has a confirmation routine.
- History Partial setting is now saved at program exit.
- Fixed bug with Shortcuts not being unhooked after they were changed.

- Added <u>PitSwap</u> feature (Ctrl+Shift+Z)
- A better History preview window . Resizes picture automatically with window resize . Highlights searched text
- Now possible to jump channels directly with Up-Down keys in the Paste-Mode.
- Added 'Partial Search' algorithm in History Tool for better, user-friendly searches.
- Added 'Reset Clipjump' option in About window to sort of uninstall Clipjump.
- History Tool is now both faster and more accurate in searching and loading.
- Performance improvements in Paste-mode navigation
- Bug fixed where the Preview Right Click option in History tool will not work.

- Made History Tool to show size of items in the window rather the Total Size in the Status bar.
- Added a new setting 'is\_duplicate\_copied' to abandon Clipjump copying duplicate consecutive clips.
- Improvements in the preview functionality , better aspect ratio adjustments.

- Now Copy Folder Path, Copy File Path and Copy File Data functions work with all file managers.
- History Tool now starts with clips sorted date-wise.
- Fixed a bug which may have stopped Clipjump to monitor Clips.

#### 29/09/2013

- fixed a bug (arised in v9.0) with COPY FILE data and the history clipsminor changes in About GUI

#### 28/09/2013

- Added a special Pit channel
- Added Size column in History tool
- Fixed a bug with newly created Clipjump Channels
- Fixed Clipboard Read error bugs

#### 23/09/2013

- Fixed a bug where items different than selected where pasted by Clipjump
- Enhancements in Settings tool

### September 20th, 2013

- "Can't Open Clipboard for writing" RANDOM BUG FIXED !
- More minor bug fixes

#### September 16th, 2013

- Make PASTE-MODE (Ctrl+V) shortcut configurable.
- New options added to copy only to system clipboard bypassing Clipjump
- Added ini setting to start Clipjump with NO-FORMATTING enabled.
- changed shortcut for insta-paste to Space
- New icons added to denote Incognito Mode and Clipboard monitoring disabled state.

### September 9th, 2013

- Minor changes in History ToolInstapaste feature updatedMinor bug fixes

#### September 7th, 2013

- Clipjump Channels now support unique names for channels.
- Image File data is now copied with Copy File Data
- Exporting and sharing clipboards is now possible via Export feature
- Fixed the slow copying in Excel and other possible apps.
- History tool is much better now, remembers user's preferences for window positions and sizes and more...
- Now possible to completely turn off History recording.
- History tool shortcut can now be changed and disabled.
- Tons of bug-fixes and minor improvements in all modules.

#### August 19th, 2013

- Added Clipjump Channels
- Total Clipjump control now possible with Clipjump Controller
- Shortcuts are configurable via the Settings editor now
- Added Copy File Data(Ctrl+Alt+D)
- Added One Time Stop feature (Alt+K)
- Added Incognito mode in Clipjump History
- History Tool Date is now sorted correctly
- Made Clipjump more resistive to by-chance Clipboard conflicts
- Pasting is now instant when needed.

### August 5th, 2013

- Removed the restriction in Non-Admin access to use Clipjump.
- Better History Tool, added ability to select Clipboard by using the Down Key in Search Box
- Added Insta-Paste feature in History Tool, press Shift+Enter on the selected clipboard item to paste it directly to the underlying window.

### 7.2

#### July 27th, 2013

- Bug fixes that originated due to Clipjump's jump to Unicode
- ClipjumpCommunicator Autohotkey lib is super-fast and superdependable now.
- ClipjumpController released to control Clipjump from external API
- Reliability improvements in Clipjump's Clipboard Monitoring
- Now History Tool shows Disk space consumption

### July 19th, 2013

- History image is now transferred directly to clipboard
- Fixed minor bugs with MS Excel
- Better (full-in-feature) History Tool now
- Better preview Window
- Professional help file included replacing Readme.txt
- Changed default Image quality to 90, allows high-quality but low in size Clipjump History images.

### June 28th, 2013

- Fixed problems with MS Excel and other office (OR Office-like) apps
- Clipjump now shows Data type in Tips. A very handy addition
- Better cosmetics now
- Better History and Settings Tool feature-wise
- Fixed minor bugs

### June 10th, 2013

- Added a GUI for editing settings.
- Added a Clipboard history tool with search to quickly locate old clipboard habitants.
- Fixed a bug when copying/moving inside zipped folders open in Explorer
- Enhanced running web links from Clipjump
- Better About Window now
- Now "[more]" symbolises more text in current clipboard
- Fixed a bug where "---> more" didnt appear when moving backwards (Ctrl + C)
- Now it's possible to suspend Clipjump's clipboard monitoring via Autohotkey script

#### **Translate**

### 5.0

# Troubleshooting

I see Japanese / Chinese characters as boxes ?

This is a uncommon <u>Windows Issue</u> and has nothing to do with Clipjump. To fix it, create a text-file named as 火 or some other chinese on the Desktop and logoff and on. You should be back to normal.

Clipboard change is not captured and <u>Settings</u> are not saved

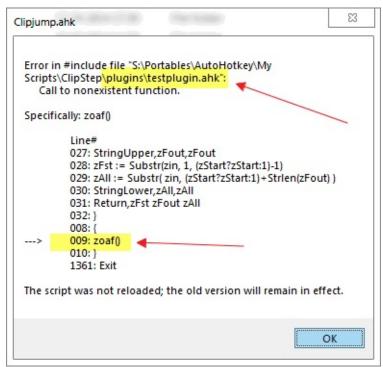
Clipjump will need to have 'write access' to the folder it is placed in. So if you are placing Clipjump in C:\ drive like *Program Files/Clipjump* you might need to **run Clipjump as administrator** to have disk rights. If you don't want to run it with administrator privileges, place it in other local drive like *D*:\ or another folder like *My Documents* where it will have rights to "atleast" write to its own folder.

Paste Mode not opening when pressing Ctrl+V

If you are seeing this bug when certain windows are active, then this happens because the active's application keyboard hook is processed before Clipjump's. To overcome this issue, **run Clipjump as administrator**. Open Clipjump.exe properties, go to Compatibility and check Run this program as administrator.

#### Error on Program startup

Majority of these errors happen because of syntax errors in a plugin. The Error message box contains all information about the error the plugin and where it is faulty in this case. In the image shown below, the first arrow shows the file (the plugin file) with the error and the second arrow shows the line number where it is faulty.



To get rid of the problem, simply remove that plugin from the plugins folder and then try contacting its developer. A screenshot of the error box will help.

#### Icons missing in Channel Organizer

As of *v11.6*, font icons are used in Channel organizer which are loaded from *icons*\*octicons*-*local.ttf*. Therefore first of all make sure that you have updated to v11.6 and the font file is present. If still there are problems, try installing the font *octicons*-*local* manually and then restarting Clipjump. Log <u>an issue</u> if you still have problems.

My plugins don't show in the Plugin Manager

If you have placed a plugin file or edited one while Clipjump was running, it will not be loaded into the program. You will have to restart Clipjump (preferrably from the System Tray Menu) to load it.

I see an error not listed on this page

<u>Contact the author</u> and that will be fixed in the next version. :-)

<u>Translate</u>

\_

# FAQ

#### Translating this HELP

A Translate Link powered by <u>**Google Translate</u>** has been added **at the bottom** of each page of this help file.</u>

Simply Click on the Translate link and you will see that page translated in your language.

How to change the language of Clipjump ?

Use the About Clipjump option in Clipjump's Tray Menu. (Link)

The Clear History button in History Tool doesn't delete the recent Clips in Ctrl+v Paste mode?

OR

I delete a clip from Ctrl+v Paste mode but it is still there in History Tool ? The <u>History Tool</u> and the <u>Paste mode</u> are two different components and there clips are independent of each other.

History Tool is meant for **history-keeping** whereas Ctrl+V Paste mode is meant to show clips contained in different channels of Clipjump. The record of all clips added to any of the channels are shown in a single window i.e.History Tool. On the other hand, clips stored in channels can be managed from the <u>Channel Organizer</u>.

I can't set a Win key shortcut in <u>Settings Editor</u>?

This is a limitation of the Hotkey control that is used. You can use the <u>ClipjumpCustom.ini</u> along with the right <u>Label</u> to create a custom shortcut for yourself.

Here is an example for creating shortcut Win+K for Hold Clip

[holdclip\_win]
bind = Win + K

run = holdClip

How do I update Clipjump?

To update Clipjump , follow these steps -

- Close Clipjump from the tray menu.
- Go to the <u>site</u> and download the latest version.
- Extract the latest version from the archive and copy its files.
- Paste the files on your current copy of Clipjump such that *new* Clipjump.exe replaces the old one.
- Run Clipjump.exe

Will updating Clipjump preserve my settings , history and clips ? Yes , you will lose nothing.

I had the copied the file/folder to Clipjump but when I paste it , nothing happens ?

• Make sure that what you are pasting is a [FILE/FOLDER] and not it's path.

- Make sure that you are pasting it in a file manager such as Windows Explorer.
- Check that the path given by Clipjump's [PASTE MODE] tip exists . This means if Clipjump's tip contains *C*:\*test.txt* , then make sure the file exists.

What is this 'cache' folder ?

The 'cache' folder stores all the clips of Clipjump. The folder was hidden in previous versions.

It contains folders with channel numbers that stores clips for that channel. eg > (clips for channel 0), (clips1 for channel 1) and so on. If you look to backup selected channels, you can copy and store clips[n], thumbs[n] and fixate[n] directory for it though it is usually recommended to **stay away from the cache directory**. Will my clips change if I use the Paste Formats ? <u>See Here</u>

I copy an image from Windows Explorer but I don't see it's preview in the Paste Mode ?

The thing you copy when you press Ctrl+C on an image file in Windows Explorer is the link to the file and not the file's data . Similarly here, you are not copying the image but the link to the image file. To copy the image to Clipjump , you have 2 ways -

• Open the image in an Image editor such as MS Paint. Select a region and then from the Right click menu, select Copy.

• Use the <u>Copy File Data</u> feature and press its shortcut while selecting the image file in Explorer or any other file manager. You will notice the Transferred to Clipjump and then the (previewed) image will be available on Clipjump.

I have disabled the *Retain Clipboard data after Restart* feature but the clips in Paste Mode are still there ?

See Here

Why do I get the warning *Clipjump is not running as administrator* ? When Clipjump, a portable application is placed in drives or folders which don't allow users to write to them , Clipjump will not be able to save its clips and thus will become of no use. To avoid such a condition, a one time warning is coded into Clipjump to aware users of the fact.

I don't want duplicate data to be stored in Clipjump . What should I do?

These are some steps you can undertake to avoid redundancy in Clipjump -

• Try cleaning Clipjump of [*File/Folder*] type data with the *Delete* [*File/Folder*] plugin.

• Whenever doing some work in which you are ought to copy duplicate/crap data to Clipboard (Clipjump) , consider using the <u>Pit</u> <u>channel</u>.

• You can also try **Disabling Clipjump** if it looks a proper option for

you.

Can History Tool only store distinct history items ?

No. The <u>History Tool</u> in Clipjump is meant to keep track of all the clips that were copied to Clipjump with their timestamps. If you are looking to free History disk consumption, try sorting the

Window with the 'Size' column.

I see the error "[The preview/path cannot be loaded]" when pasting data from zip file opened in Windows Explorer ?

This happens because the zip file contents have temporary paths . *C:/file.zip/some.txt* refers to no browsable location and and so Clipjump is not able to show the contents.

I get the 'Pit Channel not found' message when I try to use the 'PitSwap' feature ?

<u>PitSwap</u> as you may know switches to a channel named 'Pit' when it is used. As you probably have not created a channel named 'Pit', you are getting this error. Go to the channel selector GUI to create a channel named as 'Pit' and you will be OK.

How do I see a larger preview of image in the History Preview ? History Preview window automatically resizes the image contained with its dimensions . You can drag the Preview window from its edges to resize the window with the respective image. To view the maximum possible size of an image , simply maximize the Preview window. The same applies for Text Previews too.

**Translate** 

# About

Clipjump was (and is) developed by **Avi Aryan** in his spare and non-spare time. The project was started in early 2013.

### Links

### Clipjump's

Official Page Google Group Blog Github Donate Autohotkey Topic

### Author's

Web page Blog Github Twitter

Join <u>clipjump</u>'s google group for support, issues and features requests.

## Thanks

- chaz for great GUI improvements
- Ken, Luke, Morpheus, Skrell, rosto, gunner, ScottM and <u>more people at</u> <u>AHK forum</u> for pointing out critical bugs and presenting amazing suggestions.
- fump2000 for Ignore(d) Windows manager Tool and great suggestions.
- hoppfrosch for doing valuable work on documentation.
- just-me for SQLiteDb library and TooltipEx.
- nnnik for help on Clipboard formats.
- tproli for better icons.
- maksnogin for quick testing and reporting that helped Clipjump run on Russian systems.
- tic for AutoHotkey GDI+ Library
- derRaphael for his UnHTML function
- All the <u>translators</u>
- Chris and Lexikos for creating Autohotkey without which this program wouldn't have been possible.

<u>Translate</u>

# Plugins

Plugins are AutoHotkey scripts that are placed in the plugins folder in Clipjump.exe directory. They can be accessed from the <u>Plugin Manager</u> that runs with the M key in <u>Action Mode</u>.

### **Plugin Manager**

Plugin Manager is a small tool that lets you deal with currently installed plugins. You can access it with key M from Action Mode or from Tools submenu in Clipjump's Tray menu.

The UI is kept minimal and you have to right Click on a plugin row to show the various options available.

The Edit Option opens the plugin's source code. After making changes to a plugin or after adding a new plugin, restart Clipjump to make that plugin effective.

## Installing a plugin

Simply copy the .ahk file with all other dependent files in the plugins directory and **restart Clipjump**.

Note - Clipjump lists all plugins to use at ExitTime and not at runTime.
Thus Restarting Clipjump after making changes to plugins is necessary.
TIP - To disable a plugin temporarily, create a folder named (for example) *\_\_\_\_\_Disabled* in the plugins folder and move the plugin to it.

### **Running a plugin from ClipjumpCustom**

The API.runPlugin( plugin\_file\_name , param1 , param2 , param3 , ..... ) can be used to run a plugin.

If **no** parameter (param) is passed at all , then the user will be automatically asked for the needed number of parameters. Below are suitable examples -

```
[some_section]
noautorun=1
run = API.runPlugin( some_plugin.ahk )
[some_section2]
bind=Win+Alt+K
run = API.runPlugin( some_plugin.ahk , the_text , the_title )
```

### **Creating your own plugins**

If you know the AutoHotkey language, you won't take a minute starting creating your first plugins. If you are not familiar with AutoHotkey, I recommended <u>trying it</u> as it is the ever so easy and suprisingly powerful programming language. You **won't** regret. See <u>Plugins Basics</u>

## **Officially Distributed Plugins**

- 1. General
  - 1. <u>Hotstring-Paste</u> A very raw sample plugin that helps you to create Hotstrings to paste, i.e. you type a text and a clip stored by Clipjump is pasted.
  - 2. NoFormatting paste Pastes current clipboard trimming off any formatting (like HTML).
  - 3. Delete [File/Folder] Deletes clips of data type [File/Folder] from selected or all channels.
  - 4. Sync Clipjump Clipboard Updates clipjump with the current system clipboard
- 2. Paste Formats
  - 1. <u>Common Formats</u>
  - 2. Trim Formatting
  - 3. SentenceCase
- 3. External
  - 1. <u>Ini Editor</u> by rbrtryn, Added as a secondary interface to edit <u>Settings.ini</u>
  - 2. <u>Ignored Windows Manager</u> by fump2000, allows you to manage windows which will be automatically ignored by Clipjump
  - 3. History file paths cleaner Deletes copied file and folder paths from clipboard history
  - 4. Translation File Cleaner Detects invalid and duplicate keys in language files and displays them

## **More Plugins**

Third party plugins can be downloaded at the Official Addons page

<u>Translate</u>