# Richard Teunen

# General Information regarding theCoolTool.

# The development of the Cool tool was originally started by Henrik who provided much of the base of the Cool tool program. I “Inherited” the cool tool from Henrik in late January and started adding new features and fixing the tool in early February 2020. I had no knowledge of the programming language but managed to reverse the features implemented by Henrik and was able to learn the programming language pretty quickly. Release notes for the Cool tool follow below which includes Bug fixes, New features, and updates to existing features. I will then include a section for each tab within the CoolTool to explain how they work and what they are used for along with any known limitations and possible expansions.

# Known limitations

The cool tool has the following known limitations

* A screen resolution of full HD 1080X1920 must please be used with the Cool tool as some of the mouse locations are hardcoded at this stage. The system will alter the user if the screen resolution is incorrect. The system will still work but may prove inaccurate and some function will break if another resolution is used
* Sometimes when expanding nodes the cool tool works too quickly for the system that is loading in the nodes so please try to use the cool tool on a local machine and not over a weak VPN connection as it may cause issues.
* Some of the UMD and tally features still work on older versions of the vsmStudio so may need adjusting for newer versions so if you encounter any issues please let me know and I will fix them ASAP.

As the Cool Tool is a work in progress please do not use this tool in a live system unless you are 100% sure that the Cool tool will work correctly for your needs.

# Release Notes

## February 2020

* A bug that was preventing certain gadgets from being detected by the cool tool was fixed within the meta gadget builder and the system is now capable to find any node within the gadget tree if it exists.
* The meta gadget builder was upgraded to be able to handle more than 16 iterations without placing gadget parameters into the wrong metagadgets nodes. This was occurring due to the metagadgets builder not scrolling down after a gadget was dropped. The metagadgets builder can now handle any number of iterations without breaking.
* The metagadgets builder was upgraded to be able to drag multiple gadget window nodes across to the metagadgets at the same time. For instance, gain-0 to gain-16 could all be dragged across at the same time instead of doing them one by one.
* The meta gadget builder gained the ability to fill in both the alias field and the boxalias field as the parameters were dropped into the metagadgets window.
* The ability to copy and paste fields from your clipboard into the alias fields of the metagadgets window was added. Can copy from any type of text editor (excel, notepad, word, etc.), and the system will copy those in for you extremely quickly. Automatically finds the next place to paste the alias.

## March 2020

* Added a time lock to the system that would prevent the cool tool from working after a certain time to prevent the distribution of the Cool tool
* Performed an entire revamp of the Interface for ease of use. Different tabs were created and the initial start position of the Cool tool was adjusted to always be on the screen
* Developed a way to automatically change the parameter modifiers of most of the nodes within the Gadget tree. They simply set the values in a single location and then the program follows a script and immediately sets all of the parameter modifiers.
* Fixed an error with the DMV Video inputs where the SDP parameter was not detected due to whitespaces.
* Fixed an error where one of the SDP paths video inputs path was hardcoded. Adjusted the system to work with old and new values
* Added the ability to paste values into the boxalias field of the metagadgets field. Works the same as the alias copy and paste and the user can simply decide which field to paste it into.
* Revamped GUI to separate the DMV from the simple MV. This splits the tally and UMD for the DMV.

## April 2020

* Added the ability to use SDP patching with both versions 1.8 and 1.10 of the C100. It will automatically detect the version and work correctly from there.
* The SDP patching can now also support shared connections in VSM Studio by allowing the user to select which of the nodes it would like to search through for the RTP Sessions. This greatly increases the useability of the tool.
* Allowed the user to select whether the tool should delete any values in the network signals before dropping the value which greatly increases the speed of the tool
* Added automation for color correctors. By simply stating the number of color correctors, the values of the parameter modifiers, and the boxalias suffix or prefix the tool can automatically modify and drag all of the parameters from the Color correctors to the metagadgets tab as well as aliasing them and their boxaliases as they are dropped.
* Added the automation of building connections to the gadget server. The user can specify the description prefix, IP addresses, and port of the configuration, as well as the iterator for the port and the system, which will build the connections on its own. Can be repeated for as many C100s as necessary.

## Additional Additions up until January 2021

* Added the GPIO management tab to assist with automating tedious tasks within vsmStudio and also added additional keyboard shortcut options that will be discussed in the sections below.
* Added support for different types of scripts when working with the MetaGadgets builder. All of the available options and their limitations will be discussed below.

## February and March 2020

**Build Connections Tab**

* Added the ability to Subscribe to the Crossbars of new C100s using the “Subscribe to Crossbar Feature on the Build connections Tab.
* Added the ability for the Subscribe to Crossbars feature to automatically detect all of the crossbars in the user's config without the need to provide this information to theCoolTool. This feature can be disabled if necessary
* Fixed the Connectio builders ability to create both a single connection to the gadgetserver as well as the ability to create an additional connection to a different GadgetServer
* Addressed an issue where a single connection would fail under some circumstances. This should now always work correctly
* Addressed an issue where the iterator values could be incorrect when forming multiple connections.
* Added the ability to fill in this tabs information using the config file.

**SDP Patching Tab**

* Added the ability to provide the name of the first RTP session to be used if the names within the tree node have been changed from their default values. This could be further improved in the future.
* Added the ability to navigate quickly through large trees greatly reducing the time required to successfully locate nodes.
* Addressed an issue that could cause sessions to be transferred to the incorrect location if the window was not sized correctly.
* Additional fixes were made to increase the speed and efficiency of the DMV and C100 video inputs and transmitters

**Metagadget Builder**

* Added the ability to have multiple iterators in a file as well as the ability to have iterators with different start values. The iterators can be identified in the file using a special syntax or an additional coolmeta file could be used for each line. More details in the metal gadget section below
* Added the ability to define multiple parameters within a single path. This means that theCoolTool will only navigate to this path once and then drop all of the required parameters from that path before continuing. This means that the Cooltoll will not waste time navigating the tree and will instead only have to navigate the tree once to retrieve these parameters.
* Performed a test with NEPs system that required over 800 parameters to be dropped and was able to complete this task in just over 11 hours with only 8 parameters being missed due to TeamViewer connection loss. Additional speed and other optimizations were made to achieve this goal.
* Added the ability to alias and Box alias each line in a Coolmeta script and also allowed multiline drag and drops(such as scripts that drop gains 0-15 all at once) to have their own method of Aliasing.
* Moved the Parameter modifier to its own dedicated tab and also cleaned up the meta gadget builder tab.
* Addressed an issue where the paste from the Alias function would not work correctly under certain circumstances. The window was adjusted to scroll down to ensure that the user's text would be pasted into the correct areas.

**Windows Shortcut keys**

The latest addition to the CoolTool is the addition of shortcut keys. The following was completed that will always be active in the background as long as theCoolTool is open

* Added the ability to map certain windows to keys on a user's keyboard. If pressed the system will attempt to find the allocated window and will activate and display this window to the user. This is completely customizable using the Config file.
* Added the ability to control the lock icon on the master view with a single button press
* Allowed the user to control many of the properties of the GPIO window using shortcut keys such as CTRL+Shift+C will set all of the conditions to true and CTRL+Shift+F will set them to false.
* Added the ability to drag and drop between to the metagadgets window with a single button press thus saving the user extra effort.

## April 2021 Changes And Improvements

The following new changes and features were added during the month of April 2021.   
**HandMade SDP Creator**

* A completely new feature was added to assist with the creation of Handmade SDPs. The feature would receive a text file along with an Excel spreadsheet containing the required SDP along with the required values for each of the SDPs and the tool will then automatically navigate to all of the required signal path names and paste in the required SDPs. A full description of this feature will be provided in the sections below

**Metagadget Builder Improvements**

* The metagadget builder has received an additional method that will ensure that any of the containers or the dropped parameter will automatically be made visible on the screen without the need to use the mouse to scroll. This greatly increases the speed and accuracy of the tool.
* The metagadget builder now has the ability to do multiple cards at the same time where the user can specify the card to do within the gadget tree as well as specify the metagadget container where the tool will need to start from. More information on this tool can be found in the metagadget builder section
* The metagadget builder now has a built-in script builder that can be used to create metagadget scripts for you with the click of a single button.
* Various Templates and Examples uploaded to the local DropBox and onto the Yammer page
* Added the ability to extract the values from either the Alias or the Component field using a single button. More info is provided in the metagadget section.

**Keyboard shortcuts**

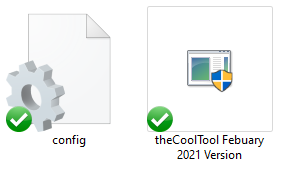
* Added a Keyboard Shortcut that will allow users to automatically lock and unlock the master matrix or any other relevant views.
* Added a keyboard Shortcut to automatically move layers in the Matrix properties window whilst ensuring that the moved items will always remain on screen for easy error spotting. An additional keyboard shortcut has been added that will automatically sort the Matrix window for you.
* Added the support to be able to use Ctrl + Tab to move between tabs on windows that did not originally support this feature(such as the master view).
* Added the ability to extract all of the layers in the Matrix Properties window using a combination of keystrokes.
* Added functionality to copy the names of metagadgets Containers as well as the ability to search through both the metagadget containers as well as the metagadget parameters
* Added the ability to easily reorder all of the layers within the Matrix properties window through the use of an Excel spreadsheet

# How to use theCoolTool

This section will instruct you on how to Install theCoolTool and how to ensure that your system is ready to make full use of all of its features.

## Installing theCoolTool

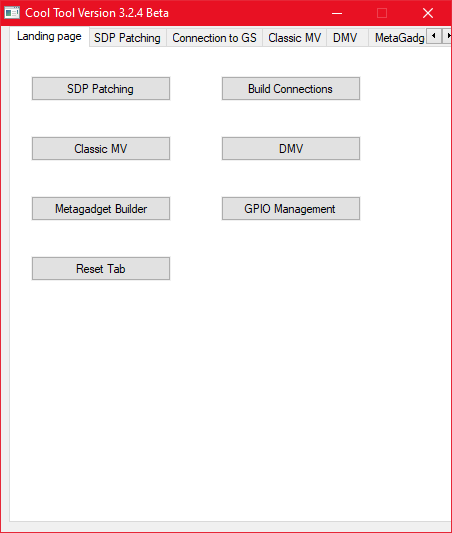
TheCoolTool is a compiled AutoIt script that has been changed into an executable file. The latest version of this executable will always be available on the Dropbox that was created for the project(Link can be found at the bottom of this documentation) under the “latest stable version” folder and this folder will always contain the latest build of theCoolTool. The executable will generally include a date to indicate when the Build was completed and will look something like this.



These are the only two files that you will require to use theCoolTool and even if you do not find the config file then the system will create one for you the first time that theCoolTool is run. The **Config File** will contain all of the user-defined keyboard shortcuts and will allow you to edit these shortcuts to suit your needs and preferences(more on this in a later section). Once you have found these files you can simply copy them to your local Desktop and run the executable to start using theCoolTool. The application will more than likely ask you for administration privileges as many of the features need to completely take over your keyboard and Mouse so you may be prompted with the following window:

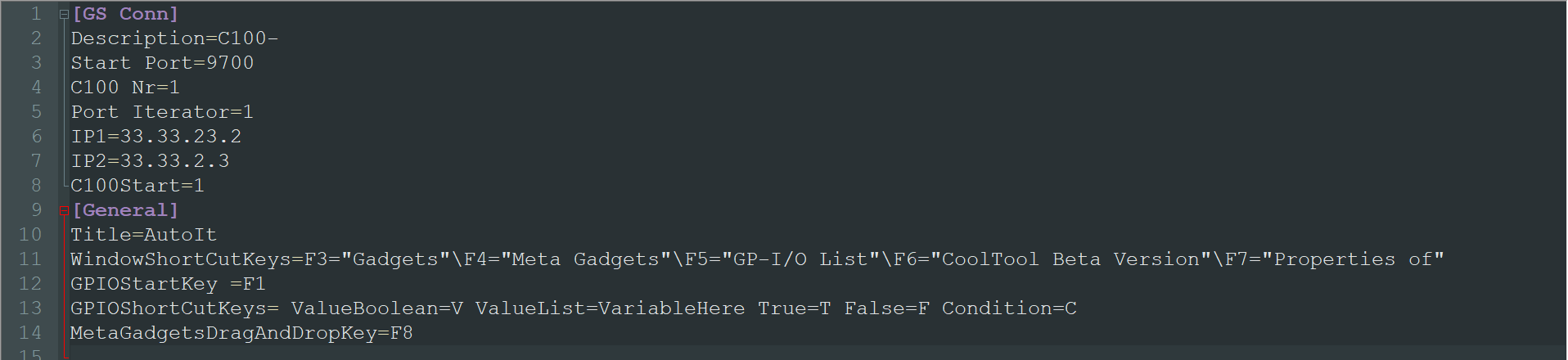


Simply click on the Yes button to allow the CoolTool access to administrator rights and you will be presented with the following home screen:



You are now ready to start using theCoolTool and I would strongly suggest creating a shortcut to the theCoolTool executable as this will make it easy to reach from any location on your PC. You are free to rename theCoolTool to anything that you want as this will not affect the performance of the tool. The next section will walk you through how to make use of the Config File to set Some of the Keyboard Shortcuts used by TheCoolTool.

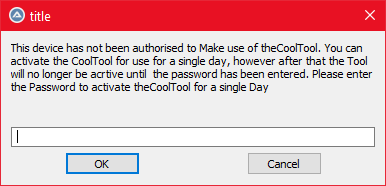
## Working with the Config File

TheCoolTool will make use of ini config file to store some critical information such as the previous states of components used by the user and also the various keyboard shortcuts that should be made available to the user. You can use any text editor to edit the config file such as Notepad++ or Visual studio code and the ini File will look something like this

The Only section that you will have to edit is the section marked as [General] and this section will generally be located at the very bottom of the Config file and there you will find various shortcut keys commands where you can assign specific keys to assign specific tasks. Full descriptions of these shortcut key commands will be provided in a separate section.

## CoolTool Licensing

A new feature has been added to theCoolTool as of May 2021 that will completely License the tool and will thus be used as a way to control who has access to the Tool. When Opening the Tool you may be presented with the following error message:



This error message means that your device has not been authenticated by any of the SSE Managers and you can either contact Richard Teunen(Developer of theCOolTool) or James Walker to gain full access to theCoolTool

## Using theCoolTool on a Client PC

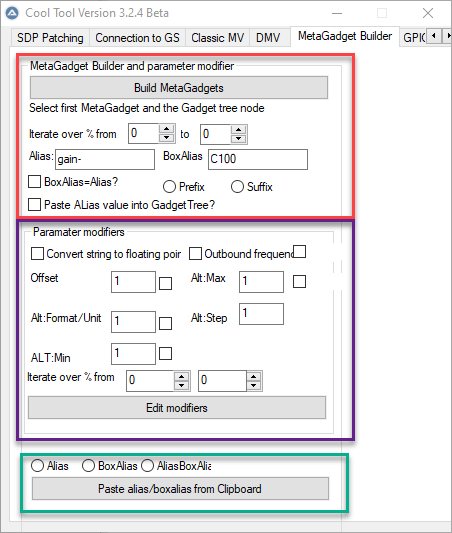
There may arise a scenario where you would want to use theCoolTool on a client's PC and in these cases, it is crucial that you do not authenticate the Tools for these devices as we don’t the device to leave the SSE team. In these cases, you can Contact Richard Teunen and he will provide you with a password that can be used to enable theCoolTool for a single day without the need to authenticate the device. This will only activate for a single day and will then be unusable until the password is entered again. For this reason please refrain from sharing the password with one another if possible. Please ensure that you have a valid internet connection during the authentication process.

## Metagadgets Builder Tab

This section will focus specifically on the meatgadget builder since this is the feature that is in high demand and could greatly ease the process of creating metagadgets. There will be various types of scripts that can be passed to the metal gadget builder so I will first explain the important components for each script and I will also provide examples for each type of script and I will explain how they work.

## The layout of the metagadgets Tab

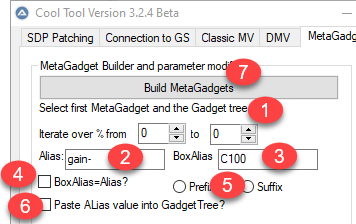
This section will discuss the Layout of the theCoolTool Metagadgets tab and will assist you with understanding what each of the components is used for. When you open theCoolTool and navigate to the metagadgets tab you will be presented with the following interface:

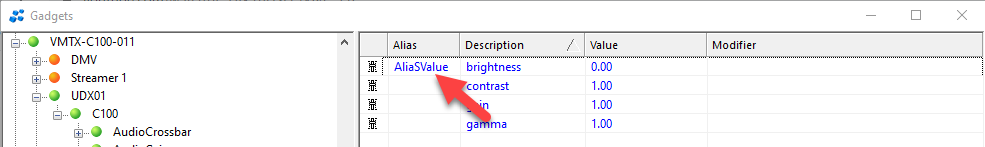


The file is split into three main categories. The first section(marked in red) will assist you with dropping and dragging parameters from the gadgets window to the metagadgets window and will make use of a script that needs to be provided to the user. The second section(marked in the purple square) can be used to fill in the various fields within the parameter modifiers and will also make use of a script to navigate through the GadgetTree. The final section marked in Green is a Paste from clipboard feature that was added to assist with pasting values into the alias or Boxalias fields of the metagadgets window. Each of these sections will be discussed below

## Metagdget Builder section explanation.

This section was specifically designed to assist with automating the task of dragging and dropping parameters from the gadgets window to the metagadgets window. The components used to achieve this goal are described below:



1. The scripts that will be used to retrieve the parameters from the gadget window have been fitted with various iterators that have been denoted with a % sign. This part of the interface will be used to collect the start and endpoints for these iterators are set to 0 and 0 by default meaning that the script will only loop once and place a value of 0 into the iterator locations.
2. With the very basic scripts, the user will only need to provide the system with a path to the parameters in the gadgets window and the system will then drop these parameters in the metagadgets window. If the user does not provide any aliases or boxaliases in the script then the system will use the values provided in components 2 and 3 to populate the Alias and BoxAlias fields. The alias will put the iterator at the end of the Alias value and the Boxalias will have a special way of being calculated based on the value selected in component 5(more info in its description).
3. Similarly to Component 2, this component will only capture the Box alias to be used if the value is not presented in the script. Component 4 and 5 will determine how the Box Alias value is constructed
4. If this checkbox is checked then the system will use the value in the Alias component(Component 2) and place it in the BoxAlias field. Thus the values and selection in components 3 and 5 will be completely ignored.
5. This component will be used to create the BoxAlias value for each parameter. When creating the BoxAlias value the user can either simply have the value placed in Component 3 be used for the BoxAlias field or the user can make use of component 5 to add the text in Component 5 to the front(prefix) or the back(suffix) of the text captured in Component 2. So for example, if the user enters Gain- in Component 2 and C100 in Component 3 and chooses the Suffix option then the BoxAlias value will be Gain-[Iterator value]C100. If the user leaves both prefix and Suffic unchecked the BoxALias will be C100 and if the prefix value is selected then the generated result will be C100gain-[iterator value]. For more complex scenarios I would suggest using the more complex scripts to avoid any possible errors.
6. If the checkbox is shown is checked then the CoolTool will take the alias that is placed in the metagadgets tab and will also paste it into the Gadgets window next to the parameter that was dragged across. An example of this is shown below. If left unchecked then this step will simply be skipped. 
7. This button can be clicked once the user is ready to begin(aka all of the iterator and alias values have been filled in) and this will prompt the user with the option to select a script that will be used to perform the drag and drops

The next section will show you how to create your scripts and will show you the various options that are available along with the ideal use cases.

### Creating Scripts for the MetaGadget Builder

In order to function correctly, the metal gadget builder requires a script that will assist it in navigating the tree in the Gadgets window. These scripts need to be placed in a .coolmeta file otherwise the system will not allow you to select the file. There are three main ways that the script can be created each with its own special use cases and limitations. These will be briefly discussed below.

**Basic Script without Alias Values**

These scripts will only contain the paths to the parameters that need to be dragged to the metagadgets window and will not contain any additional information such as Aliases or Boxaliases and these values will thus be taken from the theCoolTool GUI showed in the previous section. The user can place a % in the places where the iterator values will be located and you must also please put a: after the instance where the iterator was used as shown in the example below. All lines that start with a # are comments and will be ignored by the system.

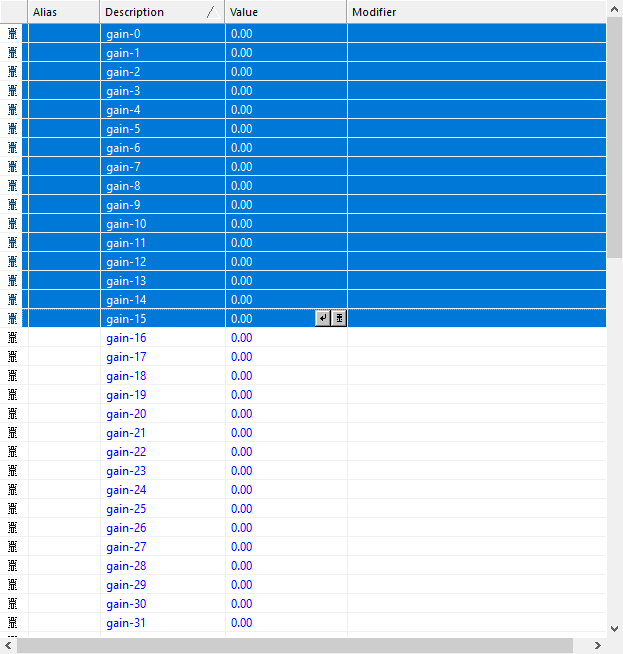
|  |
| --- |
| # Example meta gadget template file  # Each line is two parts, separated by ':'  # The first part is the series of nodes to iterate over, with '%' indicating were to iterate  # The second part is the path to the parameter within that node, either directly in the node or within a subnode  #V\_matrix/DelayHandler/delay\_handler/audio/pool[%]:outputs[0]/delay/mode\_command  V\_matrix/AudioGain/audio\_gain/gains[%]:outputs[0]/gain\_control/gain-0  V\_matrix/AudioGain/audio\_gain/gains[%]:outputs[0]/gain\_control/gain-1  V\_matrix/AudioGain/audio\_gain/gains[%]:outputs[0]/gain\_control/gain-2  # Note that the Reflect Identifier is not used, the correct device has to be selected in the Gadget Tree |

## More complex script to drag multiple parameters at once.

There are some scenarios where you may want to drag multiple gadgets from the gadget window to the meta gadget window. If you use the following script

**V\_matrix/AudioGain/audio\_gain/gains[%]:outputs[0]/gain\_control/gain-{0~15}**

Then the system will select gain 0 to gain 15 all at the same time and transfer them across to the metagadgets window. An example of how the system will select all of these paramaters is shown below. Please ensure that you use the ~ special character and the curly brackets otherwise the system will not be able to correctly identify the parameters.



This script is extremely useful if you have a lot of gain or similar properties that you want to quickly drag across. You are more than welcome to combine this type of script with simple scripts as the system will handle each line individually so you can have the following script without any issues.

|  |
| --- |
| # Example meta gadget template file  # Each line is two parts, separated by ':'  # The first part is the series of nodes to iterate over, with '%' indicating were to iterate  # The second part is the path to the parameter within that node, either directly in the node or within a subnode  #V\_matrix/DelayHandler/delay\_handler/audio/pool[%]:outputs[0]/delay/mode\_command  V\_matrix/AudioGain/audio\_gain/gains[%]:outputs[0]/gain\_control/gain-{0~15}  # Note that the Reflect Identifier is not used, the correct device has to be selected in the Gadget Tree |

**Flexible Script with individual Aliases and BoxAliases**

The final type of Scripts is the most flexible and allows the user to define their Aliases and BoxAliases. If this type of script is used then only the iterator value will be obtained from the cool tool GUI and the rest will all be obtained from the Script file. Please pay careful attention to the syntax used in the example script as it is crucial to ensuring that the coorrect values are passed to the system

|  |
| --- |
| V\_matrix/Multiviewer/multiviewer/virtual\_outputs[%]:analog\_clocks[3]/enable :Alias:"Clock01 enable"/BoxAlias:"Clock01 enable"  V\_matrix/Multiviewer/multiviewer/virtual\_outputs[%]:analog\_clocks[3]/geometry/height :Alias:"Clock01 height"/BoxAlias:"Clock01 height"  V\_matrix/Multiviewer/multiviewer/virtual\_outputs[%]:analog\_clocks[3]/geometry/posx :Alias:"Clock01 posx"/BoxAlias:"posx width"  V\_matrix/Multiviewer/multiviewer/virtual\_outputs[%]:analog\_clocks[3]/geometry/posy :Alias:"Clock01 posy"/BoxAlias:"posy width"  V\_matrix/Multiviewer/multiviewer/virtual\_outputs[%]:analog\_clocks[3]/geometry/width :Alias:"Clock01 width"/BoxAlias:"Clock01 width" |

Please ensure that you always give both the Alias and the Boxalias a value, even if it is just an empty value to ensure that the system will function correctly.

**Iterators with additional addition commands**

You also have the option to have multiple iterators to a script and could use basic addition operators to have them adjust their values during runtime.

The example below will replace the % sighns with the values provided in the CoolTool and the highlighted section will add 10 to the iterator value. Please be very careful to use the notation used below too ensure the best results are achieved.

|  |
| --- |
| C100/C100/AudioGain/levels/SRC\_AUDIO\_GAIN\_% ( +(%+10)+ ):outputs/outputs[0]/control/gain/gain[3] |

**Multiple parameters in a single path example**

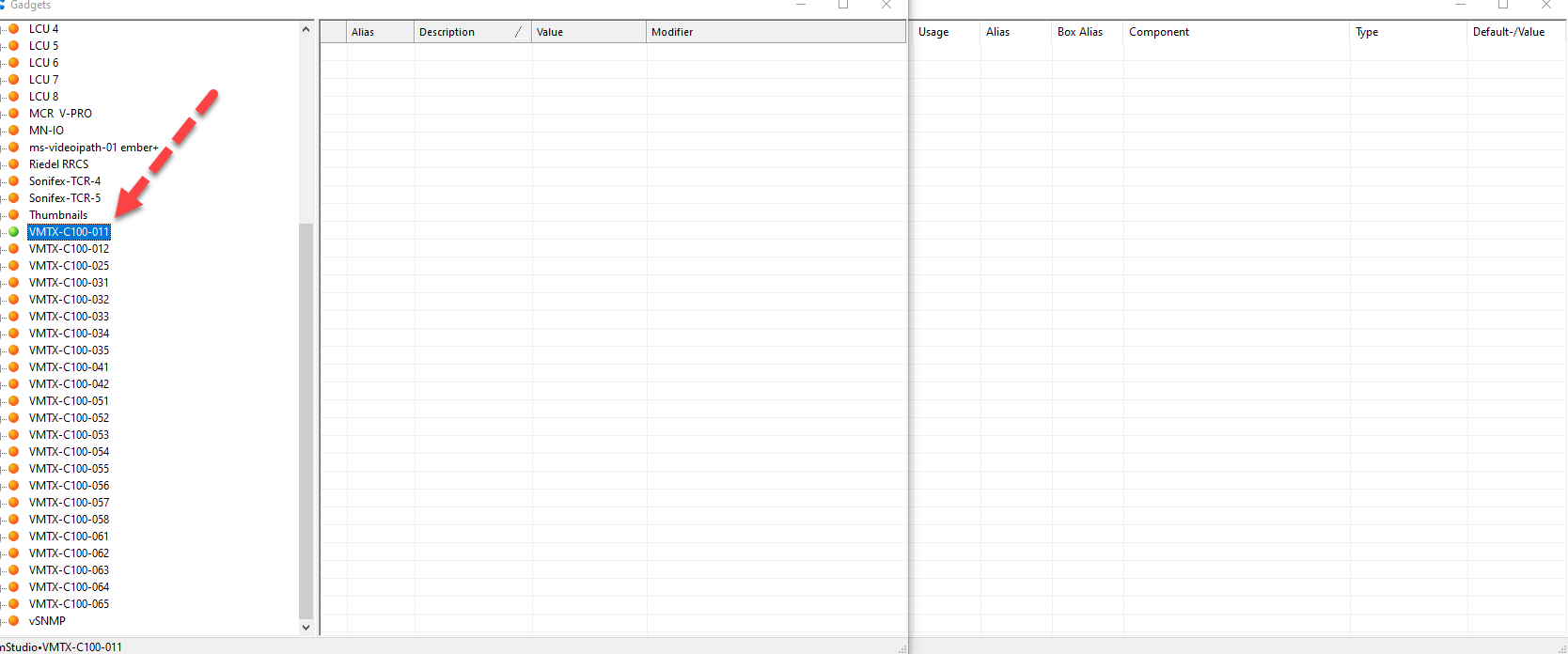
The final addition made to the metagadget builder is the ability to provide the CoolTool with multiple parameters in a single tree path. With all of the previous scripts , the CoolTool will completely reset the gadgets window tree after each line has been completed. This can waste a lot of time especially if the CooTool has to find and drag paramaters that are in the same tree path. You can tell the CoolTool to find and drag multiple parameters in the same path by using the following notation:

|  |
| --- |
| C100/C100/AudioGain/levels/SRC\_AUDIO\_GAIN\_% (+(%+10)+):outputs/outputs[0]/control/gain/gain[0],gain[2],gain[4],gain[6]  :Alias:"gain1,gain3,gain5,gain7"/BoxAlias:"gain1,gain3,gain5,gain7" |

As seen above we wish to drag gain 0,2,4 and 6 across to the metagadget window and all of these parameters are seperated using a comma. You can provide them each with their own unique Alias and BoxAlias and they are also seperated using commas.

All of the above script features can be combined in a single script and Please feel free to post any useful scripts that you have used on our CoolTool Yammer group.

Great now that you have created your script you can open up the gadget and metagadgets window within vsmstudio and select the starting node within the gadgets window and Click on the “Build meta gadgets” button in the CoolTool GUI. The node in the Gadgets window must be the last one selected before you click on the start button as shown below to ensure that the system searches for the nodes at the correct location.



The CoolTool will then ask you to confirm the start point and dropzone for the gadget properties and if they are correct then click on the yes button and select your script in the file explorer and then the system will take complete control of your system and perform the drag and drops. In order to stop the system, you can simply click on the Escape key and you should get a sound cue notifying you that the script has come to a halt. I would greatly advise making use of our more complicated scripts as this will provide the most accurate results possible and will not require the Paste from Clipboard feature to perform the Aliasing for you.

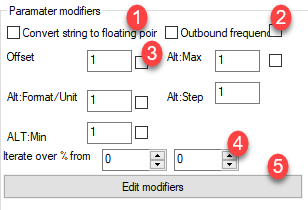
## How to use the Built-In metagadget Script Builder

The metagadget Bulder has a built-in script builder that will assist you with automatically creating the metagadget scripts that were mentioned above. All you have to do is click on the parameter in the gadgets window that you would like to add to your script(you can select 1 parameter or more than one) and then click on the F11 button and the CoolTool will automatically add the metagadget script to your clipboard for you. This can also be done by using the Keyboard Shortcut Ctrl+Shift+M as F11 is not always available to MAC users. This builder will autmatically generate the more complex scripts for the user and will populate all of the fields except the iterator that will still have to be inputted manually.

You are now fully able to use this function and if you experience any issues or bugs please let me know at [RichardTeunen2@hotmail.com](mailto:RichardTeunen2@hotmail.com) or teunenr1@lawo.com. There are also additional videos and tutorials available on the dropbox link provided at the bottom of this documentation.

## Parameter Modifier Explanation

This section of theCoolTool was added to assist users with quickly adjusting values in the parameter modifiers of multiple parameters quickly and efficiently. The following interface is used to capture information from the user and has been moved to its own tab for ease of use.



1. This component will be used to determine whether the “convert string to floating-point” checkbox must be checked or not in the parameter modifier.
2. This component will be used to determine whether the “Outbound frequency” checkbox must be checked or not in the parameter modifier.
3. This component will be used to capture the values for each of the fields from the user and a component will only be activated if the checkbox next to it has been checked. So in the example above none of the values will be set as all of the checkboxes have been left empty.
4. The scripts that will be used to retrieve the parameters from the gadget window have been fitted with various iterators that have been denoted with a % sign. This part of the interface will be used to collect the start and endpoints for these iterators are set to 0 and 0 by default meaning that the script will only loop once and place a value of 0 into the iterator locations.
5. Once you are ready to begin you can click on this button and the system will as you to provide a script for the system to use.

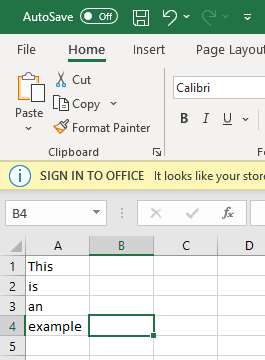
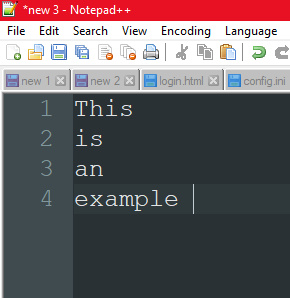
This system will only have one accepted script type at the moment and it will look something like this

|  |
| --- |
| #This value can be used with the paramter modifier function.  V\_matrix/DelayHandler/delay\_handler/audio/pool[%]:outputs[0]/delay/mode\_command |

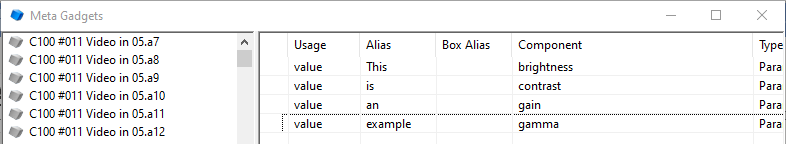
Once a script(coolmeta extension must be used) has been provided to the system it will start switching all of the values of the parameters modifiers and will notify you once it has been completed. You will lose all control of your keyboard while this is occurring, however, the script can be stopped prematurely with the escape key if needed.

## Paste from Clipboard feature

This feature was requested to assist users with quickly filling in the Aliases in the metagadgets tab by making use of the user's clipboard. This feature allows the user to set up a list for all of the values that need to be placed in the Alias or boxalias field of the metagadgets window in any text editor(simply use a new line to split the items in the list). They need only copy this list to their clipboard and the system will then “paste these values into the Alias or Boxalias fieldfd of the metagadgets window. For example, if I set up the following list in NotepAd ++ or excel :

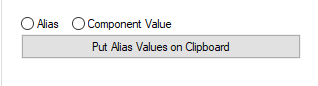
Then it will provide the following result in the Metagadgets if the Alias value is selected



This can be done for any number of parameters and it will automatically move through all of the items in the Metagadget List adding an alias/BoxAlias on each line.

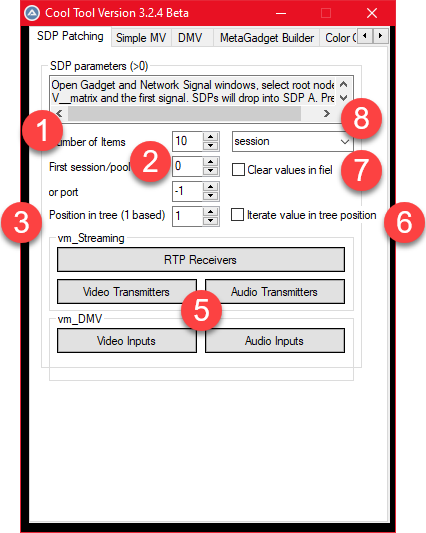
## Put Values on clipboard function

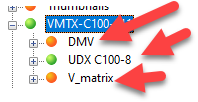
An additional function that can be used is the ability to take the Alias or components values from a metagadget window and put it on to your clipboard. This can make it easier to evaluate the Aliases and components within a metagadget window as you can paste these values into an excel spreadsheet to find any inconsistencies.



# SDP patching Tab

This tab is used for SDP patching and works with both versions 1.8 and 1.10 within vsmstudio. The GUI and its workings will be explained in this section.





1.  This component is used to say how many RTP sessions , Video transmitters, audio transmitters etc, you are trying to use. If you wish to drag across 15 RTP receivers then you would put 15 in this box
2. This component simply specifies at which RTP Session/pool the cool tool must start working from. Remember this is 0 based so if it needs to start at the 8th receiver then put a 7 here
3. This component(position in tree component) needs some explaining. Whilst doing some testing with paul we realized that sometimes more than one C100 shared a VSM Connection and thus we had a tree that looks like the tree in the image above. Usually, the cool tool would only search through the first node that it found(this case the DMV) and would simply ignore the other trees. However. I edited the tool so that you could specify in component 3 which of those nodes it should search through for the RTP sessions. So if you wanted it to look through the V\_Matrix node and get its receivers you would simply put a 3 in the box as it is the third node in that specific tree(The video I will do will explain this thoroughly)
4. I accidentally deleted 4 and did not want to redo the screenshot so ignore 4
5. These are the buttons that will determine what the cool tool will do. If you want it to Drag across video transmitters you simply click on the Video transmitters button and it will drag them across for you. The same concept goes for the other button.
6. This is a handy feature if you first want to drag across values from the first node in the tree(as shown in component 3) then the second node for the next bit, then the third node, etc. If you check this checkbox then the value in 3 will automatically increment by 1 once the tool has completed any task. This will mean that you will not have to do this manually and will save you some time. The video will also explain this nicely
7. If this checkbox is ticked then the tool will delete any fields in the network tab before it drags across a new value. If the fields are already empty then you can simply leave this box unchecked for increased speed
8. Here you simply select whether you want to drag across the session, sdp\_a or sdp\_b in the dropdown list and the program will act accordingly

# Windows(Should also work on Linux or MAC) Shortcut keys.

The CoolTool has various keyboard Shortcuts that can be accessed at any time as long as theCoolTool is open in the background. These Keyboard shortcuts have been made completely customizable and can be customized in the Config file that is created in the same directory that theCoolTool is run.

## Show Window Shortcut

Within the Config file, the user has the ability to allow key presses to focus on a certain window. The specific Line within the Config File will look something like this:

|  |
| --- |
| **WindowShortCutKeys=**F3="Gadgets"\F4="Meta Gadgets"\F5="GP-I/O List"\F6="CoolTool Beta Version"\F7="Properties of" |

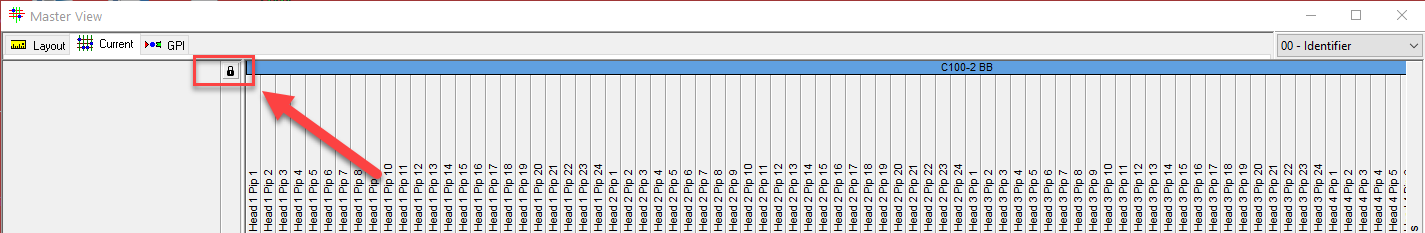
The **WindowShortCutKeys=** will automatically be generated by the system and the rest of the keys and their respective windows can be assigned by the user. Each key can be separated using a \ and the user can specify as many keys as desired. The system will do partial matches to the currently open windows on the user's device. If the window could not be found then the system will provide the user with an appropriate error message and will not open any additional windows. If several windows have the same name, then the system will activate the first window that appears alphabetically so if you wish to access a specific window please use the full name of the window.

## Metagadget Drop and Drop Shortcut

This shortcut was implemented to attempt to assist users with dropping and dragging items to the Metagadgets window. The user needs only highlight the parameter in the gadgets window that they wish to drag across and also select the correct metagadget container where the object needs to be deposited and then theCoolTool will take care of the operations for you saving both times as well as your wrist.

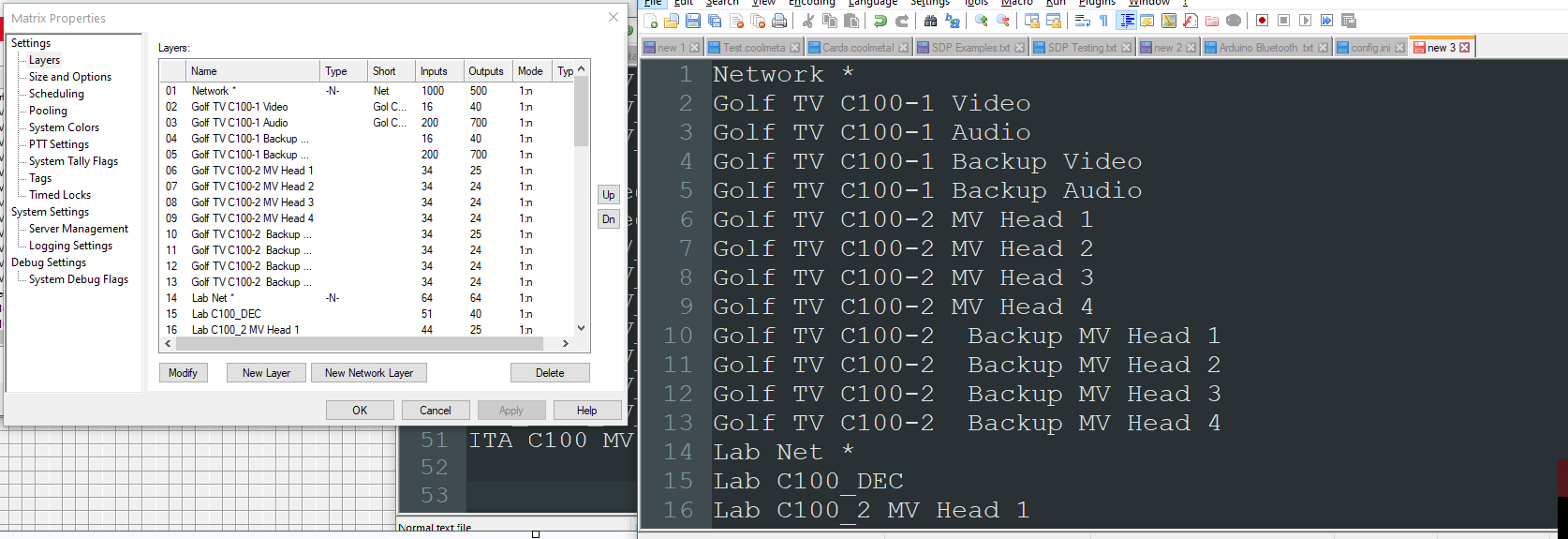
## Lock/Unlock the master View

Two additional Keyboard Shortcuts were used to Lock and Unlock the Master view or any other similar Views. This shortcut makes use of bitmaps to locate the special lock icons on the screen and will automatically click these icons for you and then return the Mouse to its original location. These keyboard shortcuts have the default key binds of F9 to lock the Matrix and F10 to unlock the Matrix, however, the user can easily change these keybinds by accessing the config file as explained in section 3. The image below displays the lock icon that the system will be searching for.



## Move Layers in the Matrix Properties window using Keyboard

Whilst working with the Layers in the matrix properties window I added Three additional Keyboard Shortcuts. The first two will move layers up or Down in the List by clicking on the ‘Up’ and ‘Dn’ buttons, however, this shortcut will ensure that the matrix list will always ensure that the selected item is visible. This means that the tool will move the item up or Down and will then automatically scroll to ensure that the layer becomes visible again. To move an item down you can simply press Ctrl+Shift+Down Keys and to move an item up you can simply press the Ctrl+Shift+Up Keys. An additional Shortcut key will extract the names of the Layer to your clipboard allowing you to easily get a list of all of the Layers currently in your configuration. The Shortcut to extract the values is Ctrl+Shift+E. The results will look something like this:



There is a special Reorder that can also be done using an Excel Spreadsheet , however, this feature will be covered in a later section.

## Find and Copy in the metagadgets Tab.

# Dropbox Link

The latest version of the Cool tool, as well as documentation, videos, and some other example scripts, can be found at the following dropbox link https://www.dropbox.com/sh/wu29srceil16mh2/AAByZ\_\_JzyOSvvaTScwL-ehJa?dl=0

# Feedback

Any feedback would be much appreciated and improvements can please be sent to Anthony Teunen and any bugs/errors to Richard Teunen (Teunenr1@lawo.com) ☺