BLACKSTAR PLUGINS

CINAPSIS

USER MANUAL

EASY ACCESS TO:
- ⊃ Objects
- ⊃ Materials
- ⊃ Textures
- ⊃ Shaders
- ⊃ Tags
- ⊃ Commands
- ⊃ Links
- ⊃ Video Post Effects
- ⊃ Descriptions

General Information

Copyright and Disclaimer

at² (former BlackStar Solutions) does not offer any warranties arising by the behaviour of the Cinapsis Plugin beyond the legal minimal warranty. This applies to any part of the product independently even if one part contravenes this disclaimer. In no case the authors are responsible for any damage or misbehaviour occurring by this product.
Copyright © 2009 by at². All rights reserved.

Licence

A single licence for this product is limited to a unique Cinema 4D licence and a single computer. The Plugin must not be transferred beyond these restrictions.

Installation

Installing BlackStar Cinapsis is similar to embedding any other Cinema 4D Plugin:

- Extract the Zip-File you received to your Cinema 4D "plugins"-folder

- On the first start-up of Cinema 4D you will be asked to enter your licence key for Cinapsis

To purchase BlackStar Cinapsis visit http://www.blackstar-solutions.de

Supported Platforms and Cinema 4D Versions

The BlackStar Cinapsis Plugin is currently available for the operating systems Microsoft Windows® and Mac OS X both 32 Bit and 64 Bit. Supported Cinema 4D Versions are 10.0 and above.

Support and Contact

For support and information beyond this manual please contact us via our support board: http://www.at2-software.com/board/ (no registration needed).

For any sales issues please contact us at: blackstar@blackstar-solutions.de

⚠ **Important:**
Concerning support requests please be sure always to add information about your operating system, your Cinema 4D version and your Cinapsis version if available. You can find your Cinapsis version in the supplied text file "version_info.txt".

Introduction and overview

Cinapsis is a workflow plugin adding something like the Mac's "Spotlight" to Cinema 4D. You can quickly find and edit any element of your scene and run commands or plugins dynamically on a specific set of items - all within an easy and light dialog.

A typical usage of Cinapsis feels like this:

have a complex scene and you're looking for some texture or shader or object - hit Alt+f, enter the first letters of the name and find a suggestion list of all the items including that in your scene.

Strike <enter> and it's selected in the Object Manager and also the Attribute Manager.
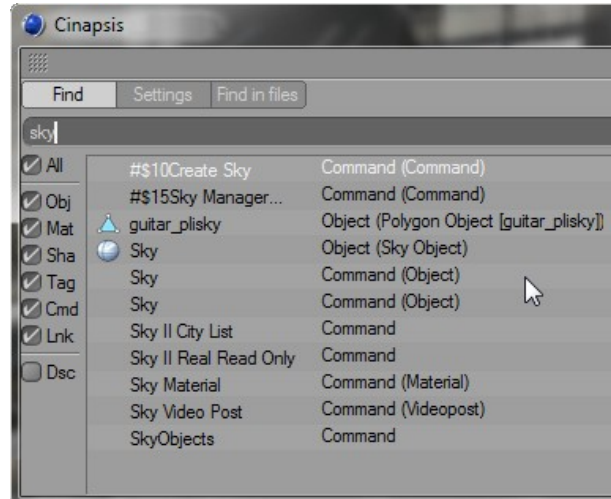
ℹ **Note:**

> *Cinapsis automatically sets Alt+f as its shortcut (or Cmd+f on the Mac) if you don't have set this to something else already. It's highly recommended to define any convenient shortcut for Cinapsis to gain a workflow speed-up. You can set shortcuts in Cinema with the Command Manager (Window -> Layout -> Command Manager).*

Controlling the search

Basically Cinapsis looks up names and displays any element which includes your given search term in its suggestion list. There are however various possibilities to adjust what Cinapsis should search for.

Only look for specific types

On the left side of the Cinapsis dialog you can define which types which should be included in the look-up:

- **Obj** – all objects in the scene as well as the object type. So typing in "light" will display all lights in your scene independent from their names because all of them are of the type "Light Object".

- **Mat** – will search for material names and also the material type.

- **Sha** – find shaders like "noise" shaders. This also includes all textures and full-length path names. So by typing in "c:\temp\" you can easily find out which textures in that folder are being used.

- **Tag** – all tags

- **Cmd** – this includes all Cinema 4D commands like "Create Sky…" which creates a new sky object, or "Remove unused materials" and such. It also includes external plugins.

- **Lnk** – find all linked items, which means all connections between items. An example would be a object "car" which is excluded by a light in its exclude list. By enabling "Lnk" and looking up "car" Cinapsis will find the light because that's where the car is referenced. That's neat to do something like a reverse search like "show me all places where xyz is linked to" or to find all instances of a specific object and such.

ℹ **Note:**

> *You can also look up scene names / documents. If you have multiple documents opened just use Cinapsis to switch between them easily. The type to be recognized in the result list is "Document".*

ℹ **Another Note:**

> *Cinapsis also looks up the used Video Post Effects in your scene, so by typing in "enDOFin" you quickly find out whether it's included in your scene or not. Of course it also is a command which would be found if you have enabled "Cmd" so watch out for the type "VideoPost".*

Regular expressions

There are simple regular expressions available to further customize the look-up:

The asterisk * stands for any character sequence so that "m*a*d" will find items "mad", "mallard" and so on. Note however that Cinapsis finds the term also if it's part of a text so you would also get an object like "Amanda" in the list although there are no asterisks as first and last character.

The pipe | symbolizes a logic OR such that "car | automobile" will find any of them.

The logic AND is represented by the character &.

Logic terms can be encapsulated using (and).

Searching for Descriptions

Enabling "**Dsc**" performs searches on descriptions. You can search for the description name like "radius" of a sphere and also any value (see image below). As soon as "Dsc" is enabled, Cinapsis does not include anything other than descriptions any more – the other check boxes like "Obj" can now be used to limit which descriptions should be included in the look up. So if only "Dsc" and "Sha" is enabled, Cinapsis will only look for description names and values in Shaders and Textures.
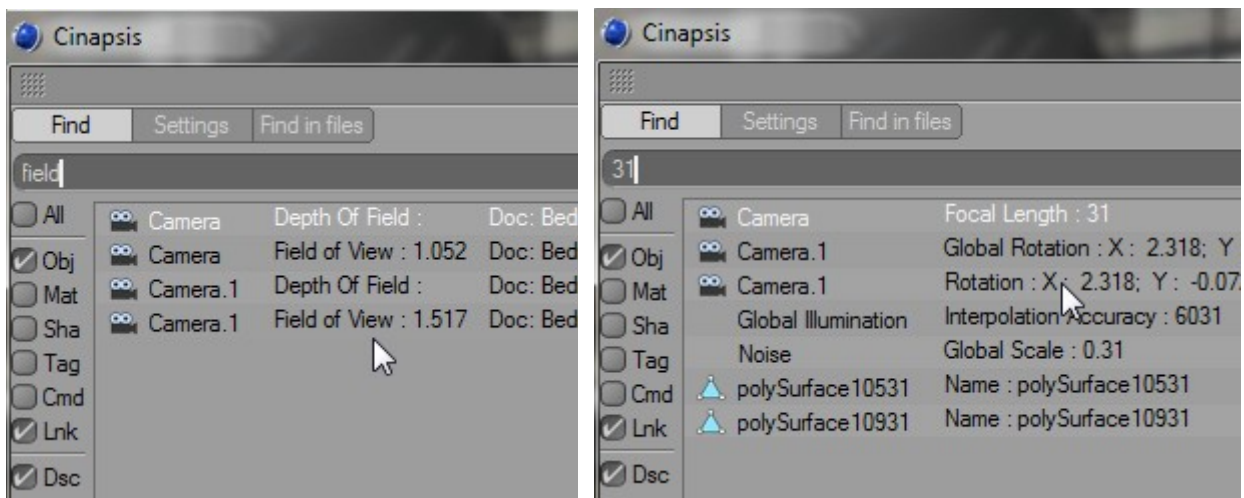
Selections and actions

Navigate through the result list with the cursor keys or the mouse. The <Enter> key or a double click will run the assigned default action (see next chapter "Settings") for the selected item. In most cases this will be selecting the element in the Object Manager and the Attribute Manager. A right click on the selection opens up a popup menu to choose any other action to process.

If you let Cinapsis find elements from all opened documents, selecting any found object or such will switch to the appropriate document automatically. This does of course not apply to Commands.

⚠ **Important:**

> Unfortunately running actions with the <Enter> key is only possible since the R11.5 update. Prior versions of C4D do not support custom key actions on tree views.

Multi-Select

You can select multiple entries of the result list. In the simple case striking <Enter> then accordingly selects all the items. Thus you have all the common properties displayed in the Attribute Manager and you now can alter a property for all items at once. That's pretty powerful, just think of finding all textures in a folder by typing in "c:\temp", doing a multi-select on all found items and then change the path property of all of them to "c:\final" at once!

Settings

Cinapsis provides a few settings to adjust. By default it finds elements of every scene which is currently open. This can be restricted such as to include only elements of the active scene.

Also you can stop Cinapsis from closing each time after an action which is of high interest if you want to dock it to your layout.

Specifying the default action

Probably most of interest is the possibility to customize what Cinapsis should do when you press <Enter> or double click an item of the result list. The functions in < > are intrinsic functions of Cinapsis. Any other action which can be chosen in the menu are C.O.F.F.E.E. scripts and can be altered or expanded freely (see next Chapter). In the Cinapsis install directory there's a folder named "scripts" with folders of each type like "Object", "Material" and "Other". Each popup-menu displays the scripts appropriate to its type on the first page and the whole folder structure including all scripts in the submenu "All actions".

"Default" resets any custom actions back to the Cinapsis interior default ones (which is <select> for everything except for Commands which get <execute>).

"Refresh" re-reads the Cinapsis scripts folder and updates the menus so that you don't need to restart the plugin when you add custom scripts.

⚠ **Important:**

> Please note that one can freely assign any script to any type, regardless whether this makes sense or not. Thus when writing or editing a script it's the duty of the scripter to take care of checking what the input is (see next Chapter for detail info).

Creating custom actions

Cinapsis can be improved freely by scripting custom actions. An action is what is being done when the user presses <Enter> or double clicks an entry of the result list. But a single action can also be called for a whole set of search results by multi-select and the right click context menu. This way you can easily perform custom tasks automated for a set of items.

If you wrote any useful scripts which you might think can be useful to others and if you are willing to share them, you are very welcome do so by posting it in our board in the Cinapsis script section: http://www.at2-software.com/board/

The scripts can be written in the Cinema 4D script language C.O.F.F.E.E.

⚠ **Important:**

> As stated above, have in mind that one can freely assign any script to any type! Thus your script **ALWAYS** has to check for types as you don't know with which input you have been called!

You can look into the existing scripts to get the basic concept of it. It's quite simple, the first argument (`_arg(1)`) is a `BaseContainer` including a `BaseLink` to every passed object. Note that it's possible to call a single script with many items – that's when it's been called with a multi-selection. The ID of the first object is 1; the others are accumulated so you can just do something like

```
while(obj = bc->GetData(id++) != NULL) { // do something }
```

to catch all of the objects. Each object can then be type-checked straight-forwardly e.g. `instanceof(BaseObject).`

The only exception are commands (or plugins) which pass the corresponding ID as parameter (also within a `BaseContainer`) to the script as this is pretty much the only thing that would make sense. Pay attention to check for `_arg(2)` not 1 for commands / plugins!

Custom script sample

Here is a simple custom script sample which can also be used as a template to start own scripts from.

```
myfunction(myArgument1, myArgument2)
{
    var bc = new(BaseContainer);
    bc = _arg(1);
    var id = 1;      // 1 is always the first ID
    var object;
    var doc = new(BaseDocument);
    doc = GetActiveDocument();
```

```
        while(bc->GetData(id) != NULL)   // iterate over all given items
        {
                object = bc->GetData(id);

                if(object != NULL)
                {
                        if (instanceof(object, BaseObject) == TRUE)
                        {
                                object->SetBit(BIT_AOBJ);
                                doc->SetActiveObject(object); // just select it
                        }
                        if (instanceof(object, BaseMaterial) == TRUE)
                        {
                                // do something with the material
                        }
                        if (instanceof(object, BaseTag) == TRUE)
                        {
                                // do something with the material
                        }
                        // Type-check other types..
                }
                id++;
        }
        EventAdd();

        // clean up
        // …
}

main()
{
        // always needed
}
```

Find elements in un-opened documents (files)

Cinapsis can also be used to search in documents not being opened: The tab "Find in files" let's you select some .c4d file directly or a whole directory (which means any c4d file in it and every subfolder) will be searched. That's pretty sweet to quickly find things out like "which scenes are using my texture on c:\temptextures"). Of course there's quite a lot to do if you let Cinapsis search in very much large scenes so this can take some time. This is also why the auto-search while typing is not active.

Needless to say running any actions on found items makes no sense and is thus not possible.

Troubleshooting

Q: I'm pressing <Enter> on some selected search result but nothing happens.

A: Unfortunately pressing <Enter> on a tree view selection is not possible in Cinema 4D versions prior to R11.5.

Q: I want Cinapsis to do something else than just selecting but I'm no C.O.F.F.E.E. scripter.

A: We're highly interested in what you think can be useful to be added to the Cinapsis script library. Please tell us what you would like to have by posting in our board http://www.at2-software.com/board/ (no registration needed) in the Cinapsis script section and we will see what we can do to get your script done.