

CATIA Version 4 Release 2

MEMORY MANAGEMENT IN AUXVIEW2, METHODOLOGY

The purpose of this document is to give some recommendations to AUXVIEW2 user who comes up against memory problems. Most of the memory problems are caused by the HLR algorithm (HLR means Hidden Lines Removal), which generates the 2D geometry. The error generally appears during STEP 4/8 of a view creation or update, an error message such as “Unable to allocate”, or “Dynamic storage cannot be allocated” is returned. This error can also appear at the very beginning of a view creation/update.

This document first gives the general considerations for the process, then it gives some recommendations for the USRENV.dcls file in order to minimize the memory allocated at the beginning of the CATIA session, and finally, it mentions some advices for the 3D geometry, and for AUXVIEW2 declaratives or parameters.

ABOUT THE PROCESS...

General points

When working on big sessions, CATIA allocates a very big amount of memory. This amount of memory can not exceed 2 Gigabytes. Thus, your operating system will be sufficient if the sum of its RAM and paging space equals 2 Gigabytes, or more (do not forget the other process running on your system !). If the RAM is important, performance will be better in term of elapse time, the ratio between elapse time and CPU time will decrease.

When operating on SUN, IRIX or HP platform, you have to dimension the paging space according to the process and the RAM (about 2 Gigabytes-RAM), and modify the system limitation with the following command :

```
ulimit -d unlimited
ulimit -a          (displays the limit)
```

The AIX operating system

On this platform, you have to specify the amount of memory segments of 256 MB, that can be used by CATIA, with the CATIA shell variable named CAT_MEM.

Example :

```
export CAT_MEM = 5
echo $CAT_MEM      (displays the value)
```

In that case, CATIA can use up to $5 \times 256 = 1280$ MB of memory.

Be careful : A CATIA interactive session can not use more than 5 segments. For bigger processes, you have to use a batch CATIA session (catutil interface), which can run on 8 segments. The batch utility for AUXVIEW2 function is called CATDRAW. See annexe "The CATDRAW batch utility".

Working in mono-model or multi-model environment ?

♣ *General points*

The multi-model use may require less memory than mono-model use. Time consuming may be bigger because multi-model links are managed. If your allocation problem becomes really critical, you may try the multi-model use, and load your 3D geometry as passive, but in return, **you have to evaluate carefully the INDEX and DATA** parameters for the USRENV.dcls file.

Remark : A passive models can not be modified, as long as it is not “activated”.

♣ *Working with passive models*

When you work with several models in a CATIA interactive session, you are able to swap these models, and change the active one. To allow this capability, CATIA store the passive models in buffer areas, which are allocated when reading passive models. This behaviour is greedy for memory, but you can avoid these allocations :

In fact, when working on your active drawing model, you do not really have to swap 3D passive models, therefore, you will be well advised to open your passive models with the option *Lock as Passive* on. With a specific declarative to be added in your USRENV.dcls file, the buffer area is not allocated.

```
CATIA.MODEL_MANAGE_OPTIMISE:STRING;  
CATIA.MODEL_MANAGE_OPTIMISE='MEMORY';
```

Recommended methodology :

- Add the previous declarative in your USRENV.dcls file
- Menu : FILE/OPEN
- Open Mode : Add Passive
- Option : Lock as passive activated
- Select your passive model(s)... OK

Remark : The option *Lock as Passive* has a second interest :

If this option is not activated, a model read in passive mode is first read as the active model, and then it is overlaid by the previous active model. The main consequence is that the MAX_ACTIVE_INDEX and MAX_ACTIVE_DATA tables have to be large enough to read a passive model.

With the option *Lock as Passive*, a passive model is directly read and overlaid, MAX_ACTIVE_INDEX and MAX_ACTIVE_DATA are no used. *See paragraph MAX_ACTIVE_INDEX and MAX_ACTIVE_DATA.*

RECOMMENDATION FOR THE USRENV.dcls FILE

Some fixed size memory blocks are allocated by CATIA at the beginning of the session, independently from the real size of the model(s) to be read. The main memory blocks are :

- MAX_ACTIVE_INDEX
- MAX_ACTIVE_DATA
- TOTAL_OVERLAY_INDEX
- GLOBAL_EXTENDED_DATA

When working on big model(s), you may have the error message MODEL FULL. This error occurs when one of the previous memory blocks is full. This has got nothing to do with allocation problems, but when you are facing allocation problems, you will be well advised to adjust those parameters, according to the model(s) to be read, in order to save static memory. See *annexe "The MAX_ACTIVE_INDEX and TOTAL_OVERLAY_INDEX tables"*.

INDEX and DATA when working on a single model

You must set the TOTAL_OVERLAY_INDEX value to zero, and inactivate the BIGACT utility. Add in your USRENV.dcls file the three following lines :

```
CATIA.MODEL_KBYTES.TOTAL_OVERLAY_INDEX = 0 ;  
CATDRAW.AUXVIEW2.BIGACT : LOGICAL ;  
CATDRAW.AUXVIEW2.BIGACT = FALSE ;
```

The MAX_ACTIVE_INDEX and MAX_ACTIVE_DATA memory blocks will be used to store :

- The existing 2D and 3D geometry, which is saved with the existing .model file
- The AUXVIEW2 views you want to create
- **Temporarily, the elements managed by AUXVIEW2 during an update/creation.**

Methodology :

- When you open a .model file, the sum of index and data necessary for the stored elements approximately corresponds to the file's size. 3 and 5 generally bound the ratio of data to index. Therefore, you can define suitable values for index and data :

$$MAX_ACTIVE_INDEX = \frac{file's\ size}{3+1} \times 1$$
$$MAX_ACTIVE_DATA = \frac{file's\ size}{5+1} \times 5$$

You have to round up these values to the nearest Kbytes

- In fact, when running CATIA, you can have a more precise idea of these values with the ERASE/PACK function which gives you the exact amount of index and data used by the active model.
- The index and data necessary to store the 2D geometry of an AUXVIEW2 view are only a few percentages of the MAX_ACTIVE_INDEX and MAX_ACTIVE_DATA tables.
- The index and data temporarily needed during an update/creation with AUXVIEW2 cannot be evaluated easily:

If you are performing a section view, or a view with breakout, AUXVIEW2 duplicates the 3D elements, and explodes the possible 3D dittos. In case of aligned section views with several cutting planes, the number of duplications may increase. In these cases, more elements will be managed, the index and data temporary needed may be more important than for a projection view.

If SAG value is too small, or if you have a lot of 3D dittos, many temporary elements will be managed by

projection software.

As a first test, you can add 20 percent to the preceding MAX_ACTIVE_INDEX and MAX_ACTIVE_DATA values.

Example :

I want to generate some projection views from a cylinder head model, the file's size is 90000 Kbytes. As a first try the parameter necessary to open the existing model are :

$$MAX_ACTIVE_INDEX = \frac{90000}{3+1} \times 1 = 22500$$

$$MAX_ACTIVE_DATA = \frac{90000}{5+1} \times 5 = 75000$$

I can get some precise values with ERASE/PACK function.

I have no ditto, the SAG value is reasonable, and that is why I add 25 percent, corresponding mainly to the temporary elements. In my USRENV.dcls file, I add :

CATIA.MODEL_KBYTES.MAX_ACTIVE_INDEX = 28000 ;

CATIA.MODEL_KBYTES.MAX_ACTIVE_DATA = 94000 ;

Remark : The BIGACT utility also works when working on a single model, but there is no interest to use it.

INDEX and DATA when working with passive model(s)

Two methodologies can be proposed, according to whether the BIGACT utility is active.

See annexe "The BIGACT utility".

❖ TOTAL_OVERLAY_INDEX with BIGACT inactivated

You have to add in your USRENV.dcls file the two lines :

```
CATDRAW.AUXVIEW2.BIGACT : LOGICAL ;  
CATDRAW.AUXVIEW2.BIGACT = FALSE ;
```

Advantages : When allocations problems are critical, this method is useful to control the memory used for the model via INDEX and DATA tables. Memory is not wasted

Disadvantages : This methodology is longer.

Methodology :

- Add up the index values given by the ERASE/PACK function, for each passive model, or use the approximate method explained in the previous paragraph with the file's size of each model, in order to evaluate these index.
- Then round this sum up to the nearest Kbytes (add 1 Kbytes), this is a suitable TOTAL_OVERLAY_INDEX value for your passive models.

Remark : The DATA necessary for the passive models is dynamically allocated by CATIA.

♣ **TOTAL_OVERLAY_INDEX with BIGACT activated**

You have to add in your USRENV.dcls file the two lines :

```
CATDRAW.AUXVIEW2.BIGACT : LOGICAL ;  
CATDRAW.AUXVIEW2.BIGACT = TRUE ;
```

Advantages : This utility may be useful when you do not want to modify your USRENV.dcls file according to the models you want to work on. This method is **flexible**.

Disadvantages : This utility generally overestimates the necessary index and data for AUXVIEW2 purposes. That is why **you may have some big additional allocations, and the available dynamic memory decreases**.

Methodology :

- The TOTAL_OVERLAY_INDEX value must be sufficient to open the passive model, consequently, it must be greater than the sum of the INDEX values given by the ERASE/PACK function, for each passive model. In fact, **you can set a bigger value, to be more flexible** with other CATIA session.
- During the process, the temporary elements managed by AUXVIEW2 can be created in the TOTAL_OVERLAY_INDEX table, if the MAX_ACTIVE_INDEX table is full. That is why the MAX_ACTIVE_INDEX value can be smaller than in the previous case (with BIGACT inactivated).

Remark : The corresponding data necessary for the temporary elements put in the TOTAL_OVERLAY_INDEX table is dynamically allocated by CATIA. **That is why this method is greedier for memory**. If CATIA is unable to allocate this data block, an error message is return at the beginning of the view creation/update.

Remark : A pre-computing in AUXVIEW2 checks the estimated amount of index and data for the current action. If those values are not suitable, AUXVIEW2 generates a warning, and suggests values for MAX_ACTIVE_INDEX, TOTAL_OVERLAY_INDEX and MAX_ACTIVE_DATA. You can carry on regardless of this warning, because the pre-computing may overvalues those parameters. The batch utility CATDRAW offers the similar pre-computing, with the option ANALYSE. *See annexe "The CATDRAW batch utility"*.

♣ **MAX_ACTIVE_INDEX and MAX_ACTIVE_DATA**

When you read a .model file in **Add Passive** mode, without the **Lock as Passive** option, CATIA temporarily needs to open the model in the active mode, and then the model is put in passive mode. The MAX_ACTIVE_INDEX and MAX_ACTIVE_DATA tables, which are dedicated to active model, are used in this case. Therefore, MAX_ACTIVE_INDEX and MAX_ACTIVE_DATA values have to be greater than the biggest model's index and data.

The Lock as Passive option is useful here, because the MAX_ACTIVE_INDEX and MAX_ACTIVE_DATA tables can be evaluated regardless the models to be read in passive mode (these tables will be smaller), on the other hand, you can not swap models : a locked passive model can not be activated.

Methodology with the Lock as Passive option:

- Use the ERASE/PACK function, or use the approximate method with the file's size, for each passive model, in order to evaluate Index and data for each model.
- As a first try, compute 25 % of the sum of the index and data for the several passive models for MAX_ACTIVE_INDEX and MAX_ACTIVE_DATA respectively.
In other words, set the MAX_ACTIVE_INDEX value to 25 % of TOTAL_OVERLAY_INDEX.

Methodology without the Lock as Passive option:

- Use the ERASE/PACK function, or use the approximate method with the file's size, for each passive model, in order to evaluate Index and data for each model.
- Then choose the biggest index and the biggest data values, **these are minimum values** for MAX_ACTIVE_INDEX and MAX_ACTIVE_DATA.
- The index and data temporarily needed during an update/creation with AUXVIEW2 cannot be evaluated easily:

If you are performing a section view, or a view with breakout, AUXVIEW2 duplicates the 3D elements, and explodes the possible 3D dittos. In case of aligned section views with several cutting planes, the number of duplications may increase. In these cases, more elements will be managed, the index and data temporary needed may be more important than for a projection view.

If SAG value is too small, or if you have a lot of 3D dittos, many temporary elements will be managed by projection software.

As a first try, keep the previous minimum values evaluated for index and data, or 25 % of the sum of the index and data for the several passive models, if this last value is bigger.

Example without the Lock as Passive option:

I have two models to open in passive mode :

Model 1:

Index = 22500 Kbytes
Data = 75000 Kbytes

Model 2:

Index = 11000 Kbytes
Data = 44000 Kbytes

The active model is new.

For TOTAL_OVERLAY_INDEX, I sum the index : $22500+11000=33500$
 For MAX_ACTIVE_INDEX, I evaluate the maximum index : $\max(22500,11000)=22500$
 For MAX_ACTIVE_DATA, I evaluate the maximum index : $\max(75000,44000)=75000$

In my USRENV.dcls file, I add :

CATIA.MODEL_KBYTES.MAX_ACTIVE_INDEX = 22500 ;
 CATIA.MODEL_KBYTES.TOTAL_OVERLAY_INDEX = 33500 ;
 CATIA.MODEL_KBYTES.MAX_ACTIVE_DATA = 75000 ;

In such a case after FILE/OPEN operations, MAX_ACTIVE_INDEX and MAX_ACTIVE_DATA table will be completely available for the temporary elements managed by AUXVIEW2, and to store the 2D generated geometry.

Remark : While the index for passive models is put in the TOTAL_OVERLAY_INDEX table, the corresponding data is dynamically allocated by CATIA.

GLOBAL EXTENDED DATA

The global extended data is used to store some space elements data for visualization purposes (HLR/HRD/SHD) and AUXVIEW2 extraction. It improves re-visualization performance at BR and AUXVIEW2 extraction, but consumes memory. You can either delete the corresponding line in the USRENV.dcls file, or evaluate the suitable value for your model(s) via ERASE / PACK function (in multi-model environment, there is only one EXTENDED_DATA working area for all the passive models). If this parameter is not set, or if the value is insufficient, CATIA will allocate dynamically the necessary amount of memory, each time a BR action or AUXVIEW2 extraction is processed, the performance may decline.

Default value :

CATIA.MODEL_KBYTES.GLOBAL_EXTENDED_DATA = 400 ;

Remark : this table is not saved with the model. It is a working table.

...OTHER PARAMETERS

♣ *MAX_PERMANENT_INDEX and MAX_PERMANENT_DATA*

The CATIA.MODEL_KBYTES.MAX_PERMANENT_INDEX and CATIA.MODEL_KBYTES.MAX_PERMANENT_DATA parameters are of no use in an AUXVIEW2 context. Set them to zero, or **delete the corresponding lines** in your USRENV.dcls file.

♣ *LONMOD_GRAPHIC and LONBUF_GRAPHIC*

The catia.LONMOD_GRAPHIC and catia.LONBUF_GRAPHIC parameters defines the maximum size, in Kbytes, of the table containing reference to graphic segments, and the amount of data CATIA displays on the physical screen. Limiting those parameters may reduce memory, and reduce the number of elements visualized.

Default values :

CATIA.LONMOD_GRAPHIC = 800 ; CATIA.LONBUF_GRAPHIC = 400 ;
--

♣ *CATIA.SESSION_MANAGER_KBYTES*

This parameter defines the memory size allocation for model management during a work session. The session manager requires about 5 Kbytes for each model manipulated during the session. The default value is 100 Kbytes, thus, in an interactive session, you can read 19 passive models, and one active model.

Default values :

CATIA.SESSION_MANAGER_KBYTES = 100 ;

RECOMMENDATIONS FOR THE 3D GEOMETRY

Working with SOLIDE IMPORT

The SOLIDE IMPORT use is very useful for critical memory problems.

Importing elements consists in declare them as external references in the resulting model. Only the Brep (Boundary Representation) is copied into the active model, thus, the index and data size necessary for the 3D model(s) may decrease, and you may be able to save static memory. It is useless to keep the original solids in the session, a link is created between the original solid, and the import.

The impact of the 3D discretization value (SAG)

As far as projection views are concerned, AUXVIEW2 produces approximate geometry, because it uses HLR algorithm. The accuracy of the draw geometry is set by the 3D discretization value (also called SAG). The recommended value is :

$$\frac{SAG}{model\ dimension} \leq 2.10^{-5}$$

SAG must be lower than the smallest geometrical parameter. For instance, with the current SAG = 0.2, if you create a cylinder whose diameter is 0.2, it will not be extracted in AUXVIEW2 context, nor be seen in HLR/SHD visualization mode.

If SAG value decreases, accuracy may be better, but time consuming, and dynamic memory amount will increase. The best value is the greatest possible value which enables you to see correctly the geometry.

Methodology :

- Evaluate the suitable SAG value, regarding your 3D geometry, or with the model-dimension rule
- Switch on 3D mode (button SP/DR)
- Go to GRAPHIC/MOD SPEC function
- Select the element you want to modify the SAG, a panel appears
- Modify the SAG value, for "Discretization values for HLR/SHD/HRD modes
- Visualization will be updated after first BR.

RECOMMENDATION FOR THE EXTRACTION, IN AUXVIEW2

The AUXVIEW2 parameters

If the previous recommendations are unable to solve your allocation problems, you can try to modify the AUXVIEW2 views parameters, in order to limit index and data tables. With the following views parameters, AUXVIEW2 generates more elements, or manages more temporary elements, that is why, in case of critical allocation problems, you may try to change those parameters.

Be aware that :

- With the hidden lines representation, much more elements are generated and managed.
- The conventional mode software for fillets is greedy for memory, and more elements are generated.
- A section view, or a view with a simple or unspec breakout, is first computed by a Boolean operation on the duplicated element(s) and exploded 3D dittos, and then, extracted by the HLR algorithm. A very large amount of temporary elements is necessary for such views, especially if the section view has several aligned elements for its cutting profile.

Remark : The section cut views process does not use the HLR algorithm. Those views are computed by a Boolean operation on the duplicated element(s), which **does not need so much dynamic memory**.

The AUXVIEW2.ALLOC.MANAGEMENT parameter

A development, realized on CATIA version 4-2-4, enables the HLR algorithm to run on a shorter amount of memory. This development is available on 4-2-3 and 4-2-2 too, via PTF.

It is dedicated to very large single SOLIDE, and is activated via an integer type declarative (to be added in the USRENV.dcls file), whose value can be set from 2 up to 12 (if the value is greater, CATIA will set it to 12, in case of other erroneous value, this functionality is not activated)

Add in your USRENV.dcls file the two lines :

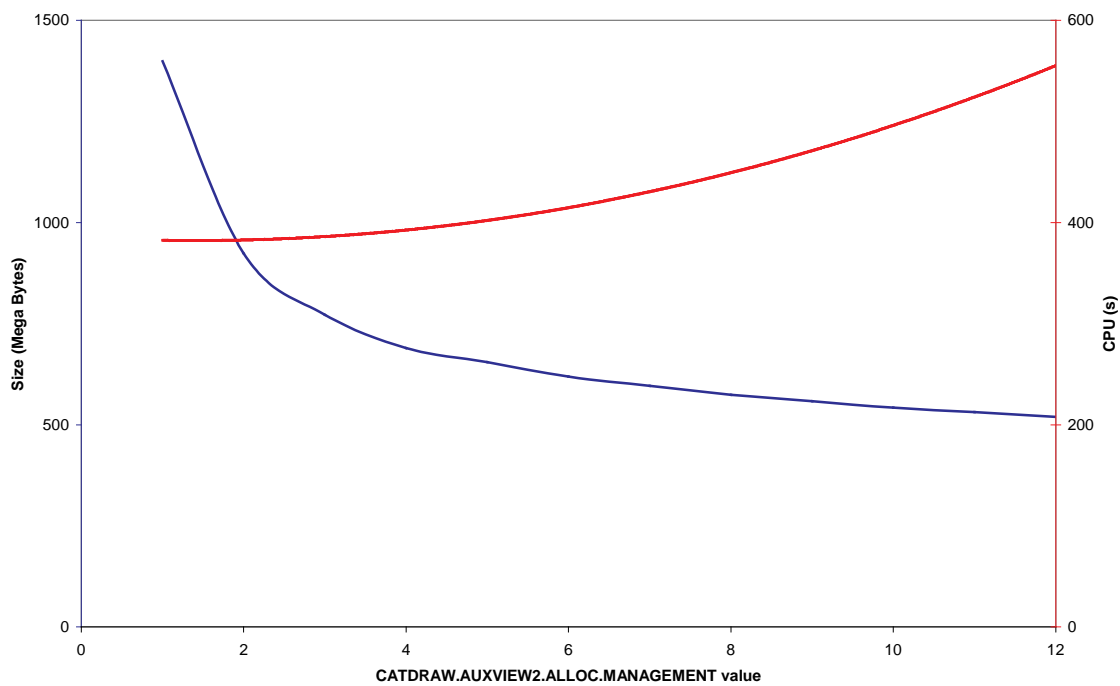
```
CATDRAW.AUXVIEW2.ALLOC.MANAGEMENT : INTEGER ;  
CATDRAW.AUXVIEW2.ALLOC.MANAGEMENT = 2 ;
```

How to choose the integer value ?

As a first test, set this value to 2. If you increase this value, the total amount of memory will decrease, but the computations will be more important, and the CPU duration may be longer. If allocation problems persist, increase it.

As an example, the following graphic shows the impact of this parameter on memory (blue curve, in Mega Bytes), and on CPU duration (red curve, in seconds). The model tested is a single SOLIDE, the .model file size is about 95 Mega Bytes.

The memory and CPU duration without this development has been represented too, it corresponds to the value 1.



Remarks : While increasing this value, you can see that the relative memory saving get lower, while CPU increases. Be aware that the memory curve is asymptotic, and that the static memory necessary for the model (about 400 Mega Bytes here : MAX_ACTIVE_INDEX, MAX_ACTIVE_DATA...) constitutes an important part of this asymptote. That is why, **despite this AUXVIEW2.ALLOC.MANAGEMENT new capability, you will be well advised to adapt your USRENV.dcls file to your model**, as explained before.

Avoid using this declarative when you work on small model : This development is of no use for small model, because memory amount is generally not very important, and the CPU may increase a lot, in compared to a big model.

ANNEXES

The MAX_ACTIVE_INDEX and TOTAL_OVERLAY_INDEX tables

General points

These two tables consist of the general descriptions of CATIA elements (types, links with other elements...), respectively for the active model, and for the passive model(s). The sum of MAX_ACTIVE_INDEX and TOTAL_OVERLAY_INDEX must be less than 128000 Kbytes. CATIA allocates one memory block for these two tables at the beginning of the session.

Default values :

CATIA.MODEL_KBYTES.MAX_ACTIVE_INDEX = 400 ; CATIA.MODEL_KBYTES.TOTAL_OVERLAY_INDEX = 2000 ;
--

The BIGACT utility

During AUXVIEW2 extraction process, the required size for index and data tables temporarily increases. The BIGACT utility allows AUXVIEW2 to use temporarily, if needed, the unused amount of the TOTAL_OVERLAY_INDEX for temporary elements, while the corresponding data is dynamically allocated in memory.

Before extraction, a pre-computing is performed, in order to evaluate the amount of index and data necessary for the process.

Here we have 3 cases :

- This amount of index < Unused MAX_ACTIVE_INDEX
The extraction can run without any problem
- This amount of index > Unused MAX_ACTIVE_INDEX + Unused TOTAL_OVERLAY_INDEX
The current index tables are not sufficient, a warning 'INSUFFICIENT MODEL SIZE' is displayed.
The /help command returns the recommended values, but you can force view creation or update.
- This amount of index > Unused MAX_ACTIVE_INDEX
And this amount of index < Unused MAX_ACTIVE_INDEX + Unused TOTAL_OVERLAY_INDEX
The extraction can run without any problem, the additional data evaluated during the pre-computing is directly allocated.

In this last case, **the additional data is allocated before the first STEP, and this amount of data is always bigger than needed**, that is why, in case of allocation problems, you will be well advised to inactive this behaviour, and to find out suitable values for the MAX_ACTIVE_INDEX, and MAX_ACTIVE_DATA.

The MAX_ACTIVE_DATA table

This table consists of the geometric descriptions of CATIA elements, **for the active model only**. The adjusted DATA necessary for the models loaded as passive is directly and dynamically allocated when loading models.

The ratio between MAX_ACTIVE_DATA and MAX_ACTIVE_INDEX depends upon the type of CATIA elements being stored. If the model contains many drafting elements, the required index may increase. If the model contains a lot of surfaces, the data value may increase. A good place to begin is to use a 3.5 : 1 ratio of data to index.

Default value :

```
CATIA.MODEL_KBYTES.MAX_ACTIVE_DATA = 1200 ;
```

The CATDRAW batch utility

CATDRAW utility allows DRAW view extraction. This means generating DRAW geometry from 3D solid elements, according to the view type (principal, section, detail, and so forth).

The UPDATE action of CATDRAW is the batch equivalent for AUXVIEW2/UPDALL : The views whose parameter "view update" is set to "allowed", are updated, and the extracted geometry is identical as when using AUXVIEW2.

Methodology to create some views with CATDRAW :

- Run an interactive CATIA session, and load your model(s)
- AUXVIEW2/DEFAULT set the "Update" parameter to "manual", in order to create some **transparent** views, without generated draw geometry
- Create your views (projection views, section views, section cut...) the space geometry will appear. In fact, no draw geometry is generated, but the view operators are created.
- For each created view, AUXVIEW2/USE/PARM, and set the "view update" to "allowed", and inactive the space transparent visualization.
- Save your model, if you work in mono-model environment, or save the whole session as a CATIA session, in case of multi-model. You can exit CATIA.
- Run catutil, and then select the CATDRAW utility.
- Give the input file (single model, or session), select the different options for the save action : REPLACE, RENAME... and then select the UPDATE action.
- Click on "EXECUTE" in order to start the process. At the end of the extraction, the result is saved in the output file previously defined. This result may be opened in a interactive CATIA session, you may create associative dimension, add dress-up...

Remark : The several possible actions via CATDRAW are :

- **UPDATE** which is the equivalent for AUXVIEW2/UPDALL
- **UPDATE EXACT** is equivalent for the AUXVIEW2/UPDALL, with AUXVIEW2/USE/PARM "projection mode" set to "wireframe". BE CAREFUL : the extracted geometry is no more associative, 3D attributes are lost (colour, layer, thickness and line type). Avoid this option.
- **DROP** is equivalent for the AUXVIEW2/USE/DROP
- **CHECK** is equivalent for the AUXVIEW2/ANALYSE/VIEWS
- **ANALYSE** gives the estimated amount of MAX_ACTIVE_INDEX, MAX_ACTIVE_DATA, and TOTAL_OVERLAY_INDEX necessary for an UPDATE action. Be aware that those results are overvalues. The BIGACT utility must be activated for this option.

Remark : On AIX system, CATDRAW is the only way to run a CATIA process bigger than 1280 MB.

Remark : When running CATDRAW, a file named CATDRAW.out is generated in your current UNIX window. It gives you information about the model to be read, error message or successful message for each view to be updated, or specific information (for CHECK and ANALYSE actions), and the index and data size of the model to be written.