

Content

1 Availability of former versions.....	1
1.1 Conversion concept.....	1
1.1.1 Preparatory measures.....	2
1.1.2 Konvertierung großer Verzeichnisse:	2
1.1.3 Error messages.....	3
1.2 Restrictions.....	5
1.2.1 Preliminary notes.....	5
1.2.2 X-Server operation.....	5
1.2.3 Length of the names of variable in the action 'variable_rule'.....	6
1.2.4 Compatibility restrictions in AQL.....	6
2 APPENDIX.....	8
2.1 Primitives / Objects.....	8
2.2 Enumeration.....	9
2.3 Type / Values.....	9
2.4 Definition / Action.....	10
2.5 Attributes.....	16

1 Availability of former versions

We receive a lot of requests for continued availability of the former versions ***Sigraph Design V3.8*** and ***I-DEAS Variant Engineering V2.0*** adapted to new operating systems.

We are now able to support these requests with our product. We called this feature '***EUKLID Design Classic***'.

Start the Converter with the following start option

```
#/cv3v4_exe -classic
```

You will get the functionality as well as Look and Feel of version V3.8 with the following characteristics and restrictions:

- Software protection by the EUKLID-Software License Manager
- .top file format of Sigraph Design V3.8
(no downward compatibility to V3.7!)
- Available on SGI, HP and Windows NT, 2000, XP
- no IDM functionality, no DM
- further restriction for the PC Version
 - Hummingbird Exceed necessary
 - no Interprocess communication,
therefore no "extern prog", no PPSI
 - no bidirectional link to I-DEAS

1.1 Conversion concept

For the conversion of old model files to EUKLID Design a separate program is started, which reads old models, converts them into the new data structure and generates new files in the appropriate directories. User elements and tables which are used within models are

converted automatically, only at their first occurrence however.
This conversion program is used only for conversion purposes.

New files are created with the following extensions:

- .mod results from model extension with .top
- .udo, .uda . results from user element extension with .top
- .udt results from table extension with .tab

Existing files (.top, .tab) will remain unchanged.

Several possibilities are available for converting older models:

- a) Calling the conversion program of the installation directory:
 `#/cv3v4_exe <file>`
 <file> = Model/User element, with or without extension '.top'.
 This method runs in batch/without graphics and is recommended if you have to convert a large number models.
- b) Calling EUKLID Design with the name of the model file or user element with or without the extension '.top' as you are used to work with the old version. This is recommended for single models.
- c) Within EUKLID Design session:
 Please start the action 'Open model' in the pulldown menu 'File' and enter '*.top' as filter in the dialog box.
 This is recommended for single models.

In the cases b. and c. the converter will be started as batch process.

In all cases the old model is converted with all of its user elements and tables. You can read a protocol of the conversion in the start window.

If the files 'xx.uda' and 'xx.udo' do already exist the user element 'xx.top' will not be converted for a second time.

1.1.1 Preparatory measures

- You must have read access for **A L L** files belonging to a model.
- You must have write access for all directories involved.
- You must have read access for the configuration file `~/user_data/design.config` of the preceeding version (entries 'li+', colors).
- Customer-specific directories '#/....' must be copied into or linked with the new '#' (e.g. customer-specific `#/sym_30b`).

1.1.2 Konvertierung großer Verzeichnisse:

If all files in a complex directory tree have to be converted we recommend to use a script as the following (UNIX):

```
#!/bin/sh
tree=..... # <== replace '.....' by directory to be processed
cd $tree; tree=`pwd`
for i in `find $tree -name '*.top' -a -type f -exec echo {} \;`
do
    cd `dirname $i`; #/cv3v4_exe `basename $i`
```

done

On Windows-NT/2000/XP we recommend to use the following AQL program:

```
#\examples\aq\convert.aql
```

Before starting the program please change your working directory to that model directory you want to convert with all its subdirectories.

1.1.3 Error messages

If compatability problems lead to conversion errors two types of relevant error messages are issued:

a) Invalid actions

Error message (example):

Undefined variable type

INVALID: variable_rule named '4231_INVALID'

- was valid in old model (sons also invalid)

This message is issued ONLY if

➡ the primitive/definition was valid in the prevoius version

➡ all parameters are valid in the new version

(Interpretation of the message see below, "Interpretation ...")

b) Geometry discrepancies

Error message (example):

GEOMETRY WRONG: circle_circlepoint named '4923_GEO<0.100'

- difference between old/new model < 0.100 mm.

Error: Precision less than 0.0001 mm.

The difference between values of old and new version is indicated in the following steps:

< 0.001, < 0.01, < 0.1, < 1.0, > 1.0 mm .

This message is not issued if the parameters were also inaccurate in the previous version.

All other messages are irrelevant for the user since they do not contribute anything to localize the basical problem. Example for such irrelevant messages:

"Objects are invalid since they were already invalid in predeceeding version or since their parameters are invalid in the new version."

These messages can help the user to modify his old or converted model in order to avoid or minimize the errors. See example below.

Identification of converted objects by name:

- If the object already had a name in the preceeding version the new created object will ALWAYS get this name.

- In .mod models, erroneous objects are marked by name according to the following scheme:

<outnumber>_INVALID
 <outnumber>_GEO<accuracy>
 (<outnumber> stands for the '.number' attribute in the preceeding version's model)

ALL erroneous objects are marked in this way, even those which are not mentioned in error messages, unless they already had a name in the old model or were already invalid in old model.

- Moreover, for user-defined objects, all objects that have no original or error diagnostic names are marked with the outnumber as their name (necessary for programming technical reasons).

ATTENTION:

The names of objects within user defined objects may not be changed as long as there are still models left unconverted that contain these user defined objects.

Example for conversion messages:

==> Starting Converter [recursion level: 0]: diag.top ==

Undefined variable type

INVALID: variable_rule named 'Radius'

- was valid in old model (sons also invalid)

GEOMETRY WRONG: circle_circlepoint named '4923_GEO<0.100'

- difference between old/new model < 0.100 mm.

GEOMETRY WRONG: circle_circlepoint named '4927_GEO<0.100'

- difference between old/new model < 0.100 mm.

No intersection of these line and circle(arc) > für den

It is not allowed to open a plane > Anwender

Plane without valid planeparts > unwichtig

(Söhne von ungültigen Objekten oder solchen, die
in der Vorgänger-Version schon ungültig waren.)

14125 objects and 12710 actions written on file diag.mod

Errors encountered during conversions:

Number of objects with geometrical precision problems .. 4

Number of invalid actions created 1

Number of actions invalid already in old model 4

Number of all primitives in the old model 18806

NOTE: actions invalid already in old model are not mentioned
in a message like 'INVALID: ' .

<== Returned from Converter [recursion level: 0]: diag.top ==>

Interpretation of these messages and error diagnosis:

- The erroneous variable rule didn't get a new name, since it already had one. The rule was: $(dh > d2) * (d2 - 0.002) + -(dh \leq d2) * dh$.
'+-' is not permitted in the new version. Thus the rule should be changed like this:

$(dh > d2) * (d2 - 0.002) - (dh \leq d2) * dh$.

- The circles cannot be converted due to extreme radii ($R=9115000000$) (Example). Perhaps you can ignore the reported geometrical discrepancy of the circles.
- The likewise inaccurate points
point_centrecircle, Name: '4924_GEO<0.100'
point_centrecircle, Name: '4928_GEO<0.100'
are basically OK, but inaccurate only due to their dependence on the circles. Therefore they are marked only by name, without a message. (It would be another error, if these points were correct in this case.)

How to use the error diagnose:

Two approaches are possible:

- a) Modify the old model and convert it again.
How to find the problem objects by using the error messages of the of the converter: if the error message was
"GEOMETRY WRONG: circle_circlepoint named '4923_GEO<0.100'",
then you have to search in the old model for a primitive with the identification number 4923. You can search for it by the following simple AQL program:

```
for i in top.list_circle do
  if i.number = 4923 then
    name (i,string (i.number))
  end
end
```

After this program has finished, the problem object has the name '4923' and thus can be selected interactively.

- b) The objects which got a name by the converter may be corrected in the new model.

Attention: If you want to convert a user element, that was saved 'with AQL program', then you have to modify both AQL program and converted user. How this can be done, can be learned from the messages during the conversion of such a user element.

1.2 Restrictions

1.2.1 Preliminary notes

All known restrictions resulting from the change in versions were consciously accepted in order to realize expanded functionality and improvements. This was the only way to keep the software up to the state of the art. Some incompatibilities were avoided at great expense in order to provide long-term protection of user investments.

1.2.2 X-Server operation

In order to obtain better graphic performance we recommend to activate the 'Backing Store' mechanism of the X-Server. The area hidden behind a dialog box is then replaced by the pixel matrix stored before the dialog box composition and no slow redraw is necessary when it is removed. However, with this X-Server setting, problems in other applications may occur.

However, we do **absolutely** recommend backing store operation of the X-Server when your workstation is used primarily for EUKLID Design.

1.2.3 Length of the names of variable in the action 'variable_rule'

The length of variable in the action 'variable_rule' in EUKLID Design is limited to 31 characters. If there should be any problem during conversion the necessary information and programs are available in our support team.

1.2.4 Compatibility restrictions in AQL

As in other programming languages, not all detail properties of AQL can be kept compatible over several versions. This affects application-defined actions (as noted in existing documentation and corresponding to the Include Libraries of other programming languages), but not the language itself (control commands, etc.). However, by means of proprietary compatibility functions, it has been attempted to hold program adaptation effort within reasonable limits. Many old AQL programs will therefore run in the current version without any changes. As a rule of thumb, the cleaner the code, the less problems must be expected. Note that the data structure created by an AQL program may differ from that of the previous version, which can result in programs no longer running logically correct.

EXAMPLE:

In the previous version the function `measure_pointpointhor (...)` created a primitive with definition type of the same name, which no longer exists as action. In order to maintain programs containing that kind of actions a compatibility function was implemented. It will create a geometrically equivalent dimension via the action 'measure_plc1plc2'. Model queries referring to 'pointpointthor' have no effect and subsequent EDIT procedures cannot work.

NOTE:

Functions which only serve to maintain compatibility are noted in the documentation. Please refrain from using these functions in new AQL programs. If such functions pose an impediment to further development, they will be removed from future versions!

Incompatibilities can also occur due to error corrections or erroneous tolerance of previous versions. For example, erroneous selectors were tolerated in several definition types, which lead to error messages in the new actions.

If parameters or properties were added they usually got the so-called 'skip' attribute. This attribute causes the parameter to be accepted if specified in an AQL program. If the parameter is not specified a standard value will be used. The action is therefore compatible and flexible at the same time. The disadvantage can be a reduced legibility of AQL programs and a reduced performance.

A further restriction is the elimination of the type "common_type". It is replaced by "Bea_action" and "Bea_object", which allows a more specific model query. Programs containing such type queries must be altered accordingly.

<i>old</i>	<i>new</i>
... if type(x) = "common_type" if type(x) = "Bea_object" ...
oder	... if type(x) = "Bea_action" ...

Note as well that in the current version other relations are generated and therefore a considerably more complex data structure is possible as with previous versions. Simple recursive review of the .sons groups is no longer sufficient to find all dependent

objects. Now the additional features such as 'effects' and 'victim parameter' must be considered as well.

In the appendix all differences are listed which could be of interest in this regard. Not only incompatibilities are listed, but also changes which lead to extended functions.

2 APPENDIX

2.1 Primitives / Objects

<i>PRIM / OBJECTS</i>	<i>description</i>		
D23bs_wcoord	new		nearly the same as coord
D23bs_blstring	new		nearly the same as string
ellips			new property axes with skip for AQL
measure	new properties font italic mode	REF_VALUE REF_OBJECT REF_OBJECT	D23bs_font_rec D23bs_angle D23bs_prop
nilprim			no more actions
plane	old old new_parameter		property 'fill_color' = plane fill color property 'color' = hatch line color 'color' = plane fill color
posball	new		position of symbol_balloon
'incompatible' means, that you have to enter the negative value of the old version in this version to get same result			
posmeas	<u>If action</u> measurepointpointhor measurepointpointvert measurepointpoint measureplc1plc2 measurechain measureradiusofsect measurediamofcircle measureangle3points measureanglesector measureangle2lines measurearclength	incompatible incompatible compatible compatible compatible compatible compatible compatible	p1.x < p2.x p1.y > p2.y Element1==line && Element2! =line posmeas exactly==0
spline	spec		new property spec with skip for AQL
symbol	new properties font italic modus height	REF_VALUE REF_OBJECT REF_OBJECT REF_OBJECT	D23bs_font_rec D23bs_angle D23bs_prop D23bs_length
text	new properties font italic mode mir	REF_VALUE REF_OBJECT REF_OBJECT REF_ENUM	D23bs_font_rec D23bs_angle D23bs_prop mir D2drw_mir

<i>PRIM / OBJECTS</i>	<i>description</i>		
	pos	REF_ENUM	D2drw_textpos_enu

2.2 Enumeration

<i>ENUMERATION</i>	<i>description</i>		
D2con_cut_relations	new	"directions" to cut relations	
D23bs_an_unit	new	allow angles given in units as lengthes	

2.3 Type / Values

<i>TYPE / VALUES</i>	<i>description</i>		
D23bs_column_index	old new	was limited due to allocated memory nested depth is unlimited	
D23bs_database_rec	new	needed for data base object.	
D23bs_tablelist	new	needed for data base tables (list of available tables)	
D2con_gi_el_id_rec	new	(value to identify a single element of a graph. instance)	
common_type	old new	Basic type for all primitives Objects are Bea_object, Actions are Bea_action	
contour_rec	new	contours don't get automatically names any more	
contour_rec	old new	contourelements created by contour_rounded got attribute 'ref2' attr. 'ref2' doesn't exist any longer	
contour_rec	old new	if contourelement is of type arc and order is false (reversed): AQL attributes 'start' and 'end' of contourelement were exchanged (can't be corrected as taken from subtype arc of contourelement which can't be inverted) works correctly now	
font_rec	old new	font was selected with fontnumber (popinteger) font is selected by fontname(font_rec)	
id_rec	old new	two references (ref1=original, ref2=next original), some internal informations only two D2con_object references (ref=effect object, ref_original=original)	
text_rec	AQL_ATTRIBUTES fontnr		
text_rec	AQL_ATTRIBUTES fontnr	does not exist any more, now use	

<i>TYPE / VALUES</i>	<i>description</i>	
	thickness	"par_font.fontname" instead real value is returned, e.g. 0.35 for text with h=3.5, medium

2.4 Definition / Action

<i>DEF / ACTION</i>	<i>description</i>	
apply_tsf		new (apply a transformation on a various number of victims; has attribute "implicit")
circle_arconcircleangles	new action name: circle_arconcirangles	Necessary because old action name was cut after ...oncircle (internal reason). -> Collution with internal truncated name of action 'circle_arconcirclepoints'.
circle_of_contour	new old	selector sel (D2con_id_rec) now points to the effect object of the contour (--> circle is always created as close as possible to the picked position). Old version didn't point to effect object but to first occurrence of original object in contour
circle_of_contour		In the old version it was possible to have line or circle in id_rec, now only circle is allowed.
circle_circlecircle circle_circlepoint circle_linecircle circle_linepoint circle_tglinecircle		circle endpoints now created as tangentpoints.
contour_parallel	new old	selector sel (D2con_id_rec) now points to the new created effect object of the contour (--> parallel contour is always created as close as possible to clicked position) Old version didn't point to effect object but to first occurrence of original object in contour
contour_parallel	old new Example	parameters le, r, and rin do not get scaled (wrong!), they do parameter le (old) = 6, scale_x = 1.5 -> drawn old: 6mm, now: 6 * 1.5 = 9 mm -> converter has to change the parameter to 6/1.5 = 4 Case 1 : scale_x = 1 OR parameter = 0 -> nothing to be done Case 2 : parameter is absolute -> converter only changes the VALUE of the parameter Case 3 : parameter not absolute -> converter REDEFINE to

<i>DEF / ACTION</i>		<i>description</i>
		length_prod = r * l1 with r (new object)=1/scale_x (number_absolute) and l1 = the old parameter =====> Editing the model has changed
contour_sketch	new old new	name of action instead of 'scetch_make_scetch' has additional parameter 'z'
contour_tracing	new	new name of action instead of 'scetch_makecont'
create_attribute	old new	strings must be set in double quotes (create_attr(e64,"\$_Version","\v1\"")) strings must be set in normal quotes (create_attr(e64,"\$_Version","v1"))
drawing_language()	obsolete	old definition
drawing_point_or_slash()	obsolete	old definition
externprog_PPSI()	obsolete	old definition
externprog_dm()	obsolete	old definition
externprog_pdm()	obsolete	old definition
externprog_pdmd()	obsolete	old definition
fase_fase fase_round		AQL-Incompatible, these definitions did not work correct in aql, because the parameter selector was missing. We do not want to be compatible with definitions which did not work! Circle endpoints now created as tangentpoints.
group_copy		- has result "Basic_group" - has add. parameter "name_ext" of type "D23bs_string" - has add. parameter "name_grp" of type "D23bs_string" - "star-alt" instead of "selection menu" - parameter "star-alt" has attribute "alt_extendible" - it is possible to edit the number of copies - it is possible to edit the transformation - substitute for "inrect_copy"
group_copymirror		- has result "Basic_group" - has add. parameter "name_ext" of type "D23bs_string" - has add. parameter "name_grp" of type "D23bs_string" - "star-alt" instead of "selection menu" - parameter "star-alt" has attribute "alt_extendible" - it is possible to edit the transformation - substitute for "inrect_mirror"

<i>DEF / ACTION</i>		<i>description</i>
Inpar inpar_ttype		These functions must be called immediately before output_save. Each object created after the first call of one of these functions will be lost.
inpar_ttype		This function uses the number of the parameter of the creation action of an object. <u>Attention</u> : This number may have changed from old to new! Be aware of using compatibility functions.
inpar_icon inpar_iconascii	old new	icon is defined as string, each character resp. bit of each character corresponds to one pixel icon is defined as a group. { "user_icon" {<int> <int> <string> ...} } or { "std_icon" [node] } or { "no_icon" }
input_dissolve		NOT implemented in this version use udo_convert_to_layer instead
input_new		<p>The parameter "name" has got an additional meaning. The identification of an UDA with this name is used, when after at least one inpar, output_save is called. Therefore it is required, that the name used with inpar is also passed to output_save.</p> <p>The following example creates an UDO and an UDA. The result of the UDA is the UDO, the inpars are the parameters of the UDA. All objects created before the first inpar are content of the UDO.</p> <pre>input_new(true,"my_user") // create objects and parameter p1,...,pn inpar(p1) ... inpar(pn) output_save("my_user")</pre> <p><u>Attention</u>: input_new cannot be called twice in a session with the same name, because after the call an UDA will be loaded and it cannot be recreated with the same id later.</p> <p>Models containing this UDA will use the new version after running the program without problems, nevertheless there are references to objects inside the UDO.</p>
(inrect_cut) --> redefine_cutall		- "star-alt" instead of "selection menu" - parameter "star-alt" has attribute

<i>DEF / ACTION</i>		<i>description</i>
		"alt_extendible"
layer_normal	old	old definition The old action will be emulated, but the edit functions of old primitive "layer" are incompatible. layer_normal (string, string, boolean) layername : string : the name of layer status : string : "active", "selectable" or "inactive" locked : boolean : true if locked
layer_create	new	new action layer_create : user = layer_create (shortstring, boolean) name : shortstring active : boolean
length_pointpoint	new	new enum dist to calculate h, x, y or z length between two points
line_of_contour	new old	selector sel (D2con_id_rec) now points to the effect object of the contour (--> line is always created as close as possible to the picked position) Old version didn't point to effect object but to first occurrence of original object in contour
line_parallellength		enum lineleftright is not compatible between old and new in case the parameter line is of type line_angleuptoline.
line_angleuptoline		The direction of the line is incompatible between old and new. This causes incompatibilities in the actions that use this line.
line_alfatgcircle		sel posi incompatible because of error in setting this selector in old version
measure_chain		This action has changed principally. The old syntax is realised as function for compatibility
model_outofselset	new	new (create a new model from a selection set) - substitute for "save part of model" (output_savesection)
move_set_to_layer layer_move_objects	old new	old definition new action
multi_duplicate		- has add. parameter "name_ext" of type "D23bs_string" - "star-alt" instead of "selection menu" - parameter "star-alt" has attribute "alt_extendible" - it is possible to edit the transformation
multi_duplicatemirror		- has add. parameter "name_ext" of type "D23bs_string"

<i>DEF / ACTION</i>		<i>description</i>
		<ul style="list-style-type: none"> - "star-alt" instead of "selection menu" - parameter "star-alt" has attribute "alt_extendible" - it is possible to edit the transformation
nilprim_inpardef(user) nilprim_nildesc ()	obsolete	
optional_z		z-setting is saved with each model individually and is available again after the loadig of the model
output_dxf()	old new	output_dxf (world, select_rect, nc_filename) xy_face : world head_selected : select_rect dxf_file : nc_filename output_dxf (level, filename, group_rec) level : level filename : filename group_rec : group_rec
output_dxfv12()	obsolete	old definition
output_outtable()	obsolete	old definition
output_plotformatted	old new	output_plotformatted (size_rec, originandscale, formattype, filename, boolc_t, filename) size : size_rec window : originandscale form : formattype name : filename sendit : boolc_t args : filename output_plotformatted (size_rec, selection_rec, format_rec, filename, boolean, dynstring) size : size_rec window : selection_rec form : format_rec name : filename sendit : boolean args : dynstring The difference is the parameter window: selection_rec instead of originandscale {{x y} scale} In AQL a peculiar scale could be given. Now this feature has become obsolete
part_ppsi()	obsolete	old definition
part_to_list		used USER must be converted into UDA's and UDO's first

<i>DEF / ACTION</i>		<i>description</i>
plane_circle		parameter 'c' changed to alt parameter of circle and ellips.
plane_sketch	new	new action
plane_tracing	new	new name of action instead of 'scetch_makeplane'
point_centrecircle		parameter circle changed to alt parameter of circle and ellips.
point_intersection		selector s incomp. : intersect line with ellips
point_intersection	old new	elements ellips and contour also allowed in first alt parameter selector > 1 allowed for tang. point selector > 1 not allowed for tang. point
point_lengthonline		selector incompatible in case line is of type line_angleuptoline.
point_pointofcircle point_pointofellips point_pointonellips		new enum abs_or_rel to create point on mirrored object relative point_pointofellips or absolute. Skip for AQL.
point_relativex1y2		new parameter an allow to create a point_relativex1y2 with angle != 0.0. Skip for AQL
popup_balloon()	obsolete	old definition
popup_cotable()	obsolete	old definition
popup_dimension()	obsolete	old definition
popup_draw()	obsolete	old definition
redefine_cutborder		- new enum parameter "cut_rel" - "star-alt" instead of "selection menu" - parameter "star-alt" has attribute "alt_extendible"
set_attribute	old new	strings must be set in double quotes (set_attr(e64,"\$_Version", "\"v1\"")) strings must be set in normal quotes (set_attr(e64,"\$_Version", "v1"))
string_fromattrib		AQL attributes like "s = string_fromattrib (obj, "AQL_attr")" not allowed. Use s = obj.AQL_attr.
symbol_balloon		parameter height is property now The old syntax is realised as function for compatibility
symbol_comment symbol_cone symbol_form symbol_raw symbol_reference symbol_section		These actions have changed principally. The old syntax is realised as function for compatibility.

<i>DEF / ACTION</i>	<i>description</i>	
symbol_spanner		
symbol_cone		less parameters now
tab_one		no longer exists. substitute is tab_file
tab_sub		rule without column_name no longer feasible
tab_instance_row		Primitive is now selected indirect, by name or in text_window. Former was Primitiv (object) had to have an object name.
tab_file (tab_one)		Changed file may be supervised (file access monitor)
tab_file (tab_one)		output is written in max. accuracy, not F10.3 This allows small accurate numbers (< 1.0)
text_absolute text_anglepos		effectpoints are created at the beginning and end of the text
text_block text_frame		an effectpoint is created at each corner of the text
text_frame	new	new action
undo_move	new	
view_define	old new	10 screen_coord (lu and rb for each view) 5 selection_rec (view_press, x_press, y_press, //lu view_release, x_release, y_release) //rb
viewdata_absolute()	obsolete	old definition
viewdate_window()	obsolete	old definition

2.5 Attributes

<i>ATTRIBUTES</i>	<i>description</i>		
Bea_object	number	old new	an integer (unique in one session only) a string of 29 bytes (unique) the attribute session_number supplies the old functionality
Bea_object	color	old new	0 means color not defined -1 means color not defined 0 means background color