

Inhaltsverzeichnis

1 Weitere Verfügbarkeit der Altversion.....	1
1.1 Konvertierungskonzept.....	1
1.1.1 Vorbereitende Maßnahmen.....	2
1.1.2 Konvertierung großer Verzeichnisse:	2
1.1.3 Fehlermeldungen.....	3
1.2 Einschränkungen.....	5
1.2.1 Vorbemerkungen.....	5
1.2.2 X-Server Betrieb.....	6
1.2.3 Länge von Variablennamen in der Aktion 'variable_rule'.....	6
1.2.4 Kompatibilitätseinschränkungen in AQL.....	6
2 ANHANG.....	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 Weitere Verfügbarkeit der Altversion

In vielen Kundenanfragen wurde der Wunsch nach weiterer Verfügbarkeit der Altversionen ***Sigraph Design V3.8*** und ***I-DEAS Variant Engineering V2.0*** unter neuen Betriebssystemen geäußert.

Wir kommen diesem Wunsch unter dem Namen '***EUKLID Design Classic***' nach.

Starten Sie dazu den Converter wie folgt mit der Startoption

```
#/cv3v4_exe -classic
```

Sie erhalten dann die Funktionalität sowie Look and Feel der Version V3.8 mit folgenden Eigenschaften / Einschränkungen:

- Lizenzierung über den EUKLID-Software License Manager
- .top Dateiformat der Sigraph Design V3.8
(nicht abwärtskompatibel zu V3.7!)
- verfügbar auf SGI, HP und Windows NT/2000/XP
- keine IDM Funktionalität, kein DM
- weitere Einschränkungen für die Windows Versionen:
 - Hummingbird Exceed notwendig
 - keine Interprozesskommunikation,
damit kein "extern prog", kein PPSI
 - keine Bidirektionalität für die I-DEAS Schnittstelle

1.1 Konvertierungskonzept

Zum Konvertieren alter Modelldateien nach EUKLID Design wird automatisch ein separates Programm gestartet, das alte Modelle liest, in die neue Datenstruktur umsetzt und neue Dateien in den entsprechenden Verzeichnissen erzeugt. In Modellen verwendete

Benutzerelemente und Tabellen werden dabei automatisch mitkonvertiert, und zwar nur beim ersten Auffinden. Dieses Konvertierungsprogramm wird nur zum Konvertieren benötigt.

Es werden neue Dateien mit den folgenden Endungen angelegt:

- .mod entsteht aus Modell mit Endung .top
- .udo, .uda . entsteht aus Benutzerelement mit Endung .top
- .udt entsteht aus Tabelle mit Endung .tab

Die bestehenden Dateien (.top, .tab) werden nicht verändert.

Zum Konvertieren alter Modelle gibt es mehrere Möglichkeiten:

- a) Aufruf des Konvertierungsprogrammes aus dem Installationsverzeichnis:
 #/*cv3v4*_exe <datei>
 <datei> = Modell/Benutzerelement, mit oder ohne Endung '.top'.
 Diese Methode läuft im Batch-Modus, ohne Graphik, und bietet sich für die Konvertierung zahlreicher Modelle an.
- b) Aufruf von EUKLID Design mit dem Namen des Modells/Benutzerelements (mit oder ohne Endung '.top'), so wie Sie es von der alten Version her kennen. Dies empfiehlt sich für einzelne Modelle.
- c) Innerhalb der EUKLID Design-Sitzung:
 Im Menü-Punkt 'Datei' die Aktion 'Modell öffnen' starten, im Dateifilter des Dialogfensters '*.top' eingeben.
 Auch dies empfiehlt sich für einzelne Modelle.

Bei b. und c. wird das Konvertierungsprogramm als Hintergrundprozess aufgerufen.

In allen Fällen wird das alte Modell mit allen seinen Benutzerelementen und Tabellen konvertiert. Im Startfenster wird die Konvertierung protokolliert.

Falls die Dateien 'xx.uda' und 'xx.udo' bereits vorhanden sind, wird das Benutzerelement 'xx.top' nicht nochmal konvertiert.

1.1.1 Vorbereitende Maßnahmen

- Auf *alle* zum Modell gehörigen Dateien muss ein lesender Zugriff möglich sein.
- Für alle beteiligten Verzeichnisse müssen Schreibrechte vorhanden sein.
- Die Konfigurationsdatei ~/user_data/design.config der alten Version muss lesbar sein (Einträge 'li+', Farben).
- Kundenspezifische Verzeichnisse '#/....' müssen in das neue '#' kopiert oder verlinkt werden (z.B. kundenspezifisches #/sym_30b).

1.1.2 Konvertierung großer Verzeichnisse:

Sollen in einem weitverzweigten Verzeichnisbaum alle Dateien konvertiert werden, dann empfiehlt sich die Anwendung einer Prozedur nach folgendem Schema (UNIX):

```
#!/bin/sh
baum=..... # <=== zu durchlaufendes Verzeichnis eintragen
cd $baum; baum=`pwd`
for i in `find $baum -name '*.top' -a -type f -exec echo {} \;
```

```
do
    cd `dirname $i`; #/cv3v4_exe `basename $i`
done
```

Für Windows-NT steht ein AQL-Programm zur Verfügung:

```
#\examples\aq1\convert.aql
```

Zur Anwendung dieses Programms wechseln Sie bitte vor dem Programmstart in das Modell-Verzeichnis, das sie mit allen seinen Unterverzeichnissen konvertieren wollen.

1.1.3 Fehlermeldungen

Kommt es wegen Kompatibilitätsproblemen zu Fehlern bei der Konvertierung, dann werden zwei Arten von Fehlermeldungen ausgegeben:

a) Ungültige Aktionen

Fehlermeldung (Beispiel):
Undefined variable type
INVALID: variable_rule named '4231_INVALID'
- was valid in old model (sons also invalid)

Eine solche Meldung erfolgt NUR, wenn

- ➡ Primitiv/Definitionsart in der Vorgänger-Version gültig waren
- ➡ alle Parameter der Aktion in der neuen Version gültig sind

(Behandlung der Meldung siehe unten, "Interpretation ...")

b) Geometrieabweichungen

Fehlermeldung (Beispiel):
GEOMETRY WRONG: circle_circlepoint named '4923_GEO<0.100'
- difference between old/new model < 0.100 mm.

Fehlerursache: Präzision kleiner als 0.0001 mm.
Die Abweichung der Values zwischen der alten und neuen Version wird in folgenden Stufen ausgegeben:
< 0.001, < 0.01, < 0.1, < 1.0, > 1.0 mm .

Eine solche Meldung erfolgt nicht, wenn die Parameter in der alten Version ebenfalls ungenau waren.

Alle übrigen Meldungen sind für den Anwender irrelevant, da sie nichts zur Eingrenzung des Grundproblems beitragen. Beispiel für solche irrelevanten Meldungen:
"Objekte sind ungültig, da sie bereits in der Vorgänger-Version ungültig waren oder da ihre Parameter in der neuen Version ungültig sind."

Solche Meldungen dienen dazu, dem Anwender zu helfen, das alte oder das konvertierte Modell so zu verändern, dass die Fehler vermieden oder minimiert werden. Siehe untenstehendes Beispiel.

Kennzeichnung der konvertierten Objekte durch Namen:

- Wenn das Objekt im alten Modell schon einen Namen hatte, wird dieser IMMER unverändert übernommen.
- In neuen Modellen werden fehlerhafte Objekte mit Namen gekennzeichnet nach folgendem Muster:

<outnumber>_INVALID
 <outnumber>_GEO<Genauigkeit>
 (dabei bedeutet <outnumber> das Attribut '.number' im alten Modell)

Es werden ALLE fehlerhaften Objekte so gekennzeichnet, auch die, die nicht in den Fehlermeldungen auftauchen, es sei denn, sie hatten bereits im alten Modell einen Namen oder waren bereits ungültig.

- In benutzerdefinierten Objekten (UDO) werden darüberhinaus alle Objekte, die keinen Original- und Fehlerdiagnose-Namen haben, mit der Outnumber als Namen gekennzeichnet (nötig aus programmtechnischen Gründen).

ACHTUNG:

Die Namen von Objekten in UDOs dürfen solange nicht verändert werden, wie noch nicht alle Modelle konvertiert sind, die diese UDOs enthalten.

Beispiel für Meldungen des Konverters:

==> 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 dieser Meldungen und Fehlerdiagnose:

- Die fehlerhafte Aktion 'variable_rule' erhielt keinen Namen, da sie schon einen hatte. Die Formel lautete: $(dh > d2) * (d2 - 0.002) + -(dh \leq d2) * dh$. '+' ist in der neuen Version nicht mehr zulässig. Deshalb sollte die Formel wie folgt umgeformt werden:
 $(dh > d2) * (d2 - 0.002) - (dh \leq d2) * dh$.
- Die Kreise können wegen extremer Radien ($R = 911500000\text{mm}$) nicht exakt konvertiert werden (Beispiel). Der auftretende Fehler von 0.1 mm kann aber möglicherweise toleriert werden.
- Die ebenfalls ungenauen Punkte
point_centrecircle, Name: '4924_GEO<0.100'
point_centrecircle, Name: '4928_GEO<0.100'
sind in sich in Ordnung, werden nur ungenau durch ihre Abhängigkeit von den Kreisen.
- Deshalb werden sie nur durch Namen gekennzeichnet, es erfolgt keine Meldung. (Es wäre ein weiterer Fehler, wenn diese Punkte in diesem Fall richtig wären.)

Aus der Fehlerdiagnose resultierende Vorgehensweise:

Grundsätzlich sind zwei Vorgehensweisen möglich:

- a) Das alte Modell wird modifiziert und erneut konvertiert. Auffinden der Problem-Primitive anhand der Fehlermeldungen des Konverters. Lautet die Fehlermeldung beispielsweise
"GEOMETRY WRONG: circle_circlepoint named '4923_GEO<0.100'",
dann ist im alten Modell nach dem Primitiv zu suchen, das eine Identifikationsnummer 4923 besitzt. Dazu könnte folgendes einfache AQL-Programm dienen:

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

Nach Ablauf dieses Programms hat das Problem-Primitiv den Namen '4923' und kann somit interaktiv selektiert werden.

- b) Die im neuen Modell durch den Konverter mit Namen gekennzeichneten Objekte werden korrigiert.

Achtung: Wird ein Benutzerelement konvertiert, das 'mit AQL-Programm' gespeichert wurde, dann müssen das AQL-Programm und das konvertierte Benutzerelement nachgearbeitet werden. Wie dies geschehen kann, geht aus den Meldungen beim Konvertieren dieses Benutzerelementes hervor.

1.2 Einschränkungen

1.2.1 Vorbemerkungen

Alle aufgeführten Einschränkungen, die sich aus dem Versionsumstieg ergeben, wurden bewusst in Kauf genommen, um Funktionserweiterungen und Verbesserungen durchführen zu können. Nur um diesen Preis ist es möglich, Software auf den Stand der Technik zu halten. Zum Teil wurden Inkompatibilitäten mit sehr hohem Aufwand vermieden, um Anwenderinvestitionen langfristig zu sichern.

1.2.2 X-Server Betrieb

Um eine bessere Grafik-Performance zu erhalten, wird empfohlen, den X-Server im "Backing Store" Modus zu betreiben. Das verdeckte Feld hinter einem Dialogfenster wird dann durch die vor dem Aufbau des Dialogfensters gespeicherte Pixelmatrix ersetzt und es muss kein (zeitaufwendiger) Redraw erfolgen, wenn das Dialogfenster entfernt wird. Im Zusammenhang mit dieser X-Server-Einstellung kann es zu Problemen mit anderen Applikationen kommen. Für den hauptsächlichen Einsatz Ihrer Workstation für EUKLID Design empfehlen wir aber unbedingt den "Backing Store" Betrieb des X-Servers.

1.2.3 Länge von Variablennamen in der Aktion 'variable_rule'

Die Länge der Variablen in der Aktion 'variable_rule' ist in EUKLID Design auf 31 Zeichen beschränkt. Sollten bei der Konvertierung Probleme auftreten, stehen im Support notwendige Informationen und Programme bereit.

1.2.4 Kompatibilitätseinschränkungen in AQL

Wie auch in anderen Programmiersprachen sind nicht alle Detaileigenschaften aus AQL über die Versionen voll kompatibel zu halten. Dies betrifft, wie auch schon in der bestehenden Dokumentation eingeschränkt, die Applikationsdefinierten Aktionen (das entspricht den Include Bibliotheken anderer Programmiersprachen), nicht die Sprache selbst (Steueranweisungen etc.). Über eigene Kompatibilitätsfunktionen wurde aber versucht, den Aufwand für die Programmanpassungen in Grenzen zu halten. Viele alte AQL Programme werden also ohne jede Änderung ablaufen.

Hier gilt: je sauberer programmiert wurde, desto weniger Probleme sind zu erwarten. Beachten Sie bitte, dass dadurch nicht die gleiche Datenstruktur wie in Vorgängerversionen entsteht, was dazu führen kann, dass Programme logisch nicht mehr richtig ablaufen.

BEISPIEL:

Die Funktion 'measure_pointpointhor (...)' erzeugte in den Vorgängerversionen ein Primitiv gleichnamiger Definitionsart, die es als Aktion nun nicht mehr gibt. Um Programme mit solchen Inhalten ablauffähig zu halten, wurde eine Kompatibilitätsfunktion implementiert, die ein geometrisch gleichwertiges Mass, allerdings durch die Aktion 'measure_plc1plc2' erzeugt. Dies bewirkt, dass Modellabfragen, die sich auf "pointpointhor" beziehen, leer ausgehen bzw. Anschliessende EDIT-Vorgänge nicht mehr ausgeführt werden können.

HINWEIS:

Funktionen die NUR der Erhaltung der Kompatibilität dienen, sind als solche in der Dokumentation gekennzeichnet.

Verwenden Sie bitte diese Funktionen nicht in neuen AQL-Programmen.

Stehen solche Funktionen der Weiterentwicklung im Wege, werden sie in folgenden Versionen entfernt!

Inkompatibilitäten können auch durch Fehlerbehebungen oder fehlerhafte Toleranz der Vorgängerversionen entstehen. So wurden z.B. Fehlerhafte Selektoren in Definitionsarten toleriert, die in den neuen Aktionen zu Fehlerausgaben führen.

Wurden Parameter oder Properties zugefügt, so wurden diese meist mit dem sogenannten "skip" Attribut versehen. Dieses Attribut bewirkt, dass der Parameter bei Vorhandensein in einem AQL-Programm akzeptiert wird, bei Fehlen mit einem Standardwert versehen wird. Die Aktion wird damit kompatibel und flexibel zugleich. Der Nachteil besteht evtl. in einer schlechteren Lesbarkeit der AQL-Programme und in einer verschlechterten Performance.

Eine weitere Einschränkung stellt der Entfall des Typs "common_type" dar. Dieser wird durch "Bea_action" und "Bea_object" ersetzt, was eine differenziertere Modellabfrage erlaubt. Programme, die solche Typabfragen enthalten, müssen entsprechend geändert werden.

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

Beachten Sie auch, dass in der vorliegenden Version andere Relationen aufgebaut werden, und damit eine wesentlich komplexere Datenstruktur möglich ist als in den Vorgängerversionen.

Ein einfaches rekursives Durchlaufen der '.sons' Gruppen reicht nicht mehr aus, um alle abhängigen Objekte ausfindig zu machen. Hier ist es nun nötig, auch die neuen Möglichkeiten wie "Effekte" und "victim parameter" zu bedenken.

Im Anhang finden Sie alle interessanten Differenzen, die in diesem Zusammenhang von Interesse sein könnten. Hier sind nicht nur Inkompatibilitäten, sondern auch Änderungen aufgeführt, die zu Funktionserweiterungen führen.

2 ANHANG

2.1 Primitives / Objects

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

2.2 Enumeration

<i>ENUMERATION</i>	<i>Beschreibung</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>Beschreibung</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>Beschreibung</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>Beschreibung</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_tglineline		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>Beschreibung</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_attrib(e64,"\$_Version","\v1\\")) strings must be set in normal quotes (create_attrib(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>Beschreibung</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>Beschreibung</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>Beschreibung</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

DEF / ACTION		Beschreibung
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_attrib(e64,"\$_Version", "\"v1\"")) strings must be set in normal quotes (set_attrib(e64,"\$_Version", "v1"))
string_fromattrib		AQL attributes like "s = string_fromattrib (obj, "AQL_attrib")" not allowed. Use s = obj.AQL_attrib.
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>Beschreibung</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>Beschreibung</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