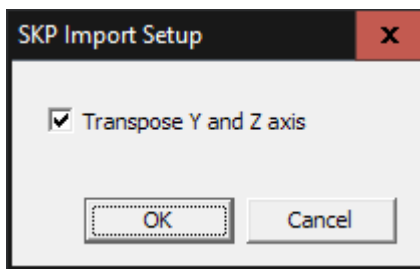


DesignCAD 3D Max 2016 Notities bij de publicatie

Hierna vindt U een gedetailleerd overzicht van de nieuwigheden in deze versie.

Import/Export Kenmerken

SketchUp Import – DesignCAD 2016 kan nu SketchUp tekeningen importeren. SketchUp versies tot SketchUp 2015 worden momenteel ondersteund. Bij het importeren van SketchUp tekeningen wordt elk afzonderlijk Solid object geïmporteerd als een aparte Solid oppervlakte entiteit in DesignCAD.



Het SketchUp importeren dialogvenster bevat een optie om Y en Z om te zetten. Indien aangevinkt, is het mogelijk in DesignCAD geïmporteerde SketchUp-objecten "rechttop" (goeie zijde naar boven) weer te geven. Deze optie is beschikbaar omdat DesignCAD de Y-as gebruikt als verticale as, terwijl SketchUp de Z-as gebruikt voor de verticale as. Als deze optie afgevinkt wordt gelaten dan zullen de ingevoerde SketchUp elementen op hun kant liggen in plaats van rechttop.

Invoeg beheer (Insert Manager)

Bestand / invoeg beheer (File/Insert Manager) is een nieuwe opdracht die het mogelijk maakt voor de gebruiker om alle blokken, symbolen en beeldbestanden die verwijzing naar of ingebed zijn in de tekening, te bekijken en te beheren.

Wanneer de "invoeg beheer" (Insert Manager) opdracht wordt uitgevoerd, verschijnt er een dialogvenster met drie tabbladen, een voor Blokken, afbeeldingen en symbolen.

Blokken (Blocks)

Het tabblad **Blokken** toont een visuele lijst van alle blokken die aanwezig zijn in de tekening. Wanneer een enkel blok oplicht door erop te klikken dan wordt een groter voorbeeld onder de lijst weergegeven. Onder de grotere weergave staat meer informatie over het blok: de naam, het ID, en een Referentie Aantal.

Rechts klikken op de lijst van de symbool iconen toont een context menu voor het sorteren van de blokken. Blokken kunnen worden gesorteerd op naam, ID, of Referentie Aantal.

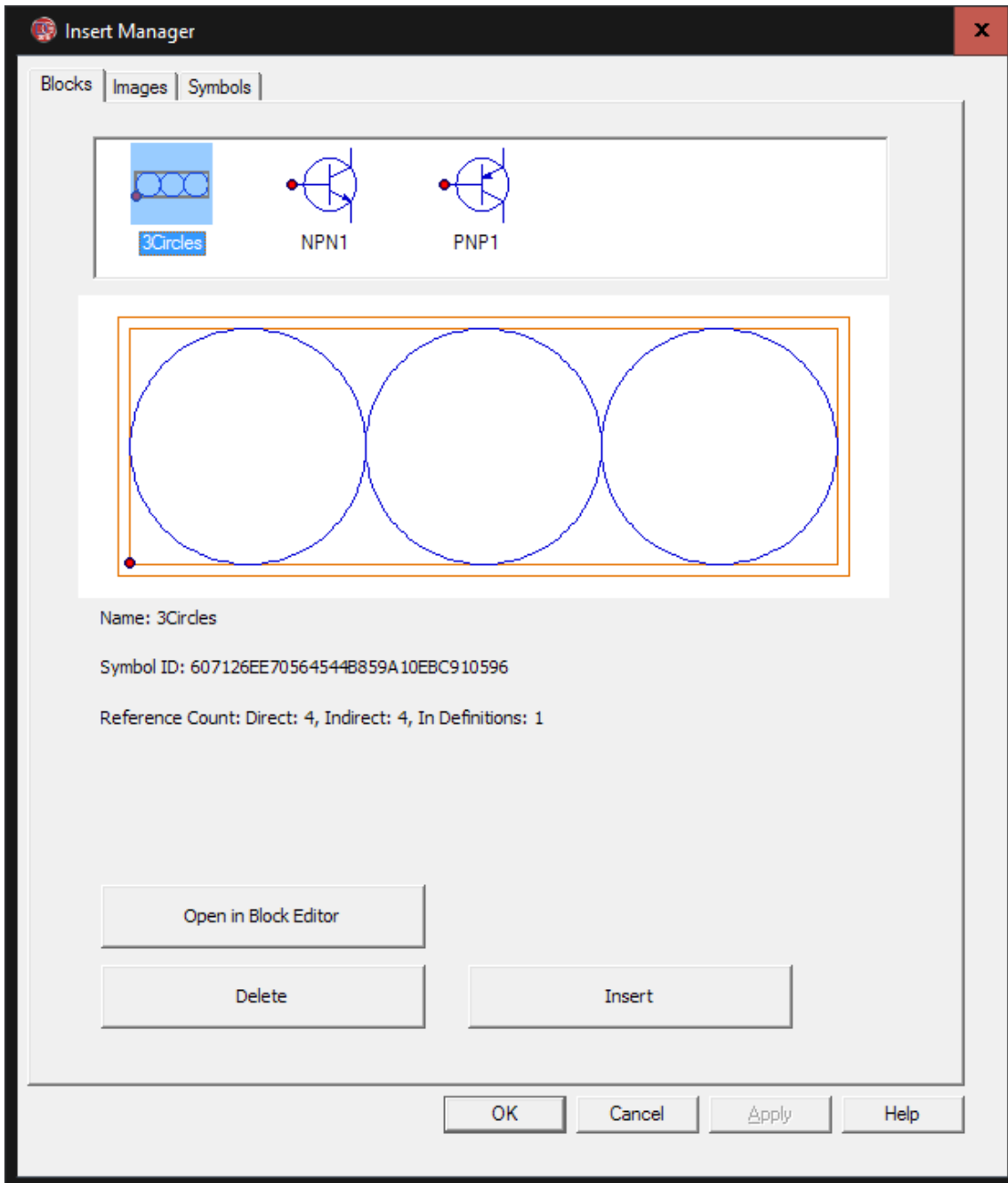
Rechts klikken op de **naam** lijn geeft u de mogelijkheid deze naam te kopiëren naar het klembord.

Rechts klikken op de **symbool ID** lijn geeft u de mogelijkheid dit ID te kopiëren naar het klembord (Bijvoorbeeld voor gebruik in een macro of COM-programma).

De "referentie aantal" lijn toont drie verschillende waarden: **directe**, **indirecte** en **in definities**. De **directe** telling is het aantal keren dat het blok in de tekening als een alleenstaand blok voorkomt (d.w.z. niet als onderdeel van een ander blok of symbool).

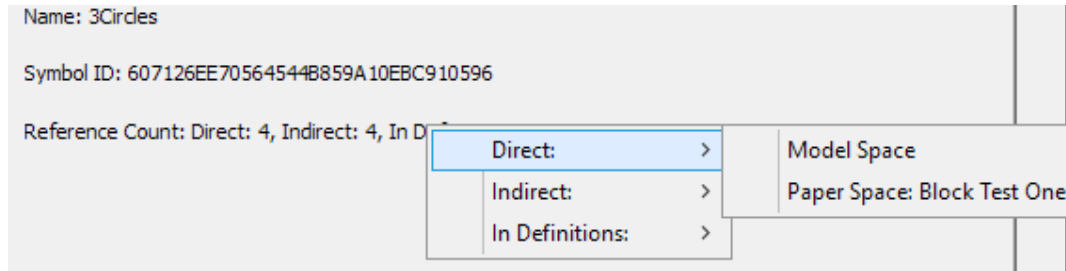
De **indirecte telling** geeft aan hoe vaak het blok is begrepen in andere blokken of symbolen die zijn opgenomen in de tekening.

De "in definities" telling geeft aan hoeveel blok- of symbool definities behoren tot het blok als onderdeel van die definitie.



Rechts klikken op deze **referentie** lijn voor een meer gedetailleerd beeld van hoe het blok wordt gebruikt, een context menu verschijnt met dezelfde categorieën: **directe**, **indirecte**, en **in definities**.

De uitbreiding van de **Direct** categorie zal aangeven of het blok in gebruik is door het model ruimte zicht (Model Space view) van de tekening en / of door een of meer papierkader indelingen (Paper Space layouts). De uitbreiding van de **indirecte** categorie zal laten zien welke symbolen en / of blokken in actief gebruikt zijn in dit blok. Uitbreiding van de categorie "**in definities**" zal aangeven welke symbolen en / of blokken dit blok omvatten, ongeacht of er al of niet actieve verwijzingen zijn naar deze symbolen / blokken.



Het tabblad blokken van het invoeg beheer (Insert Manager) biedt drie bewerkingen: **1-Open in Blok beheer**, **2-verwijderen** en **3-invoegen**.

Door te klikken op de knop **Invoegen** zal het commando Blok invoegen (Block Insert command) geactiveerd zijn en het dialoogvenster Blok invoegen tonen met het gemarkeerde blok reeds geselecteerd.

Door te klikken op de knop **Verwijderen** kunt u alle instanties van het gemarkeerde blok verwijderen uit de tekening, evenals de onderliggende blok definitie.

Als u op de **Open in blok beheer** (Open in Block Editor) knop klikt word er een speciaal venster met een beperkt menu en een opdrachten lijst geopend. In het blok beheer kunt u bijna alle gewenste wijzigingen in de blokdefinitie maken, zonder afgeleid te worden van andere tekening entiteiten. Zie hieronder blok beheer.

[\(Block Editor\)](#)

Afbeeldingen (Images)

Het **Afbeeldingen** tabblad van het invoegen beheer is vrij gelijkaardig aan het tabblad Blokken, met alle rasterafbeeldingen die zijn toegevoegd aan de tekening. De meeste opties voor Afbeeldingen zijn dezelfde als voor Blokken. Beelden bevatten ook een "**pad veld**", een "**index veld**" en een "**Geïntegreerd veld**", dit zijn extra sorteermogelijkheden als je met de rechtermuisknop op de lijst met miniatures aan de top klikt.

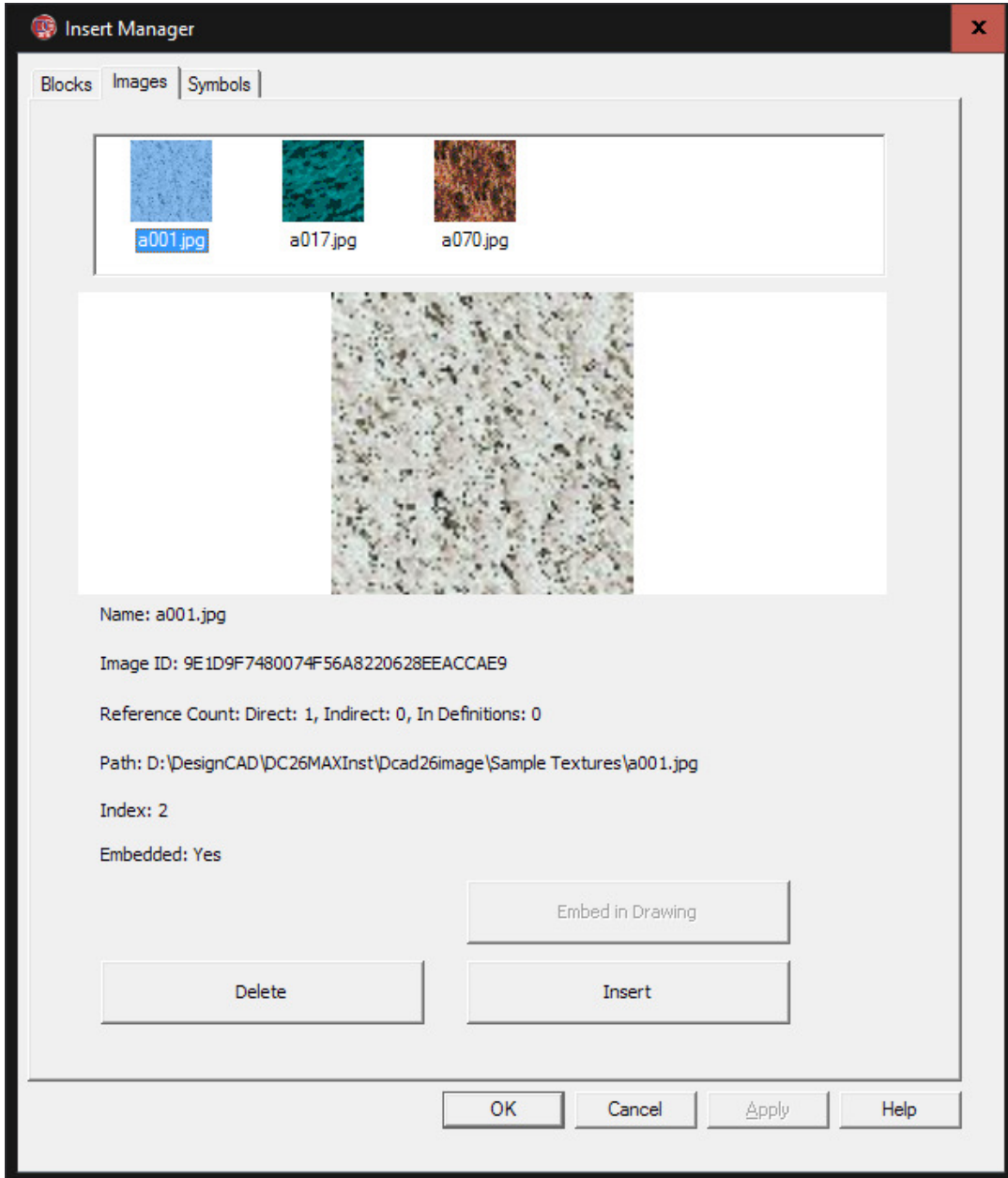
Het **Pad** veld geeft het volledige pad en de bestandsnaam van het oorspronkelijke beeld. Voor ingesloten afbeeldingen kan het originele bestand niet meer bestaan op de locatie waarnaar wordt verwezen. Door rechtsklik op het veld "**Pad**" kunt u het volledige pad en de bestandsnaam kopiëren naar het klembord voor later gebruik.

Het **index** veld geeft de index van het beeld in de lijst met afbeeldingen, d.w.z. de relatieve volgorde waarin het beeld origineel werd toegevoegd aan de tekening.

Het **geïntegreerd** veld geeft aan of het beeld geheel is opgeslagen in de tekening (Yes) of alleen een verwijzing is naar een extern bestand (No). Blokken worden altijd ingebed, dus deze knop is verborgen op de Blokken pagina (Blocks)

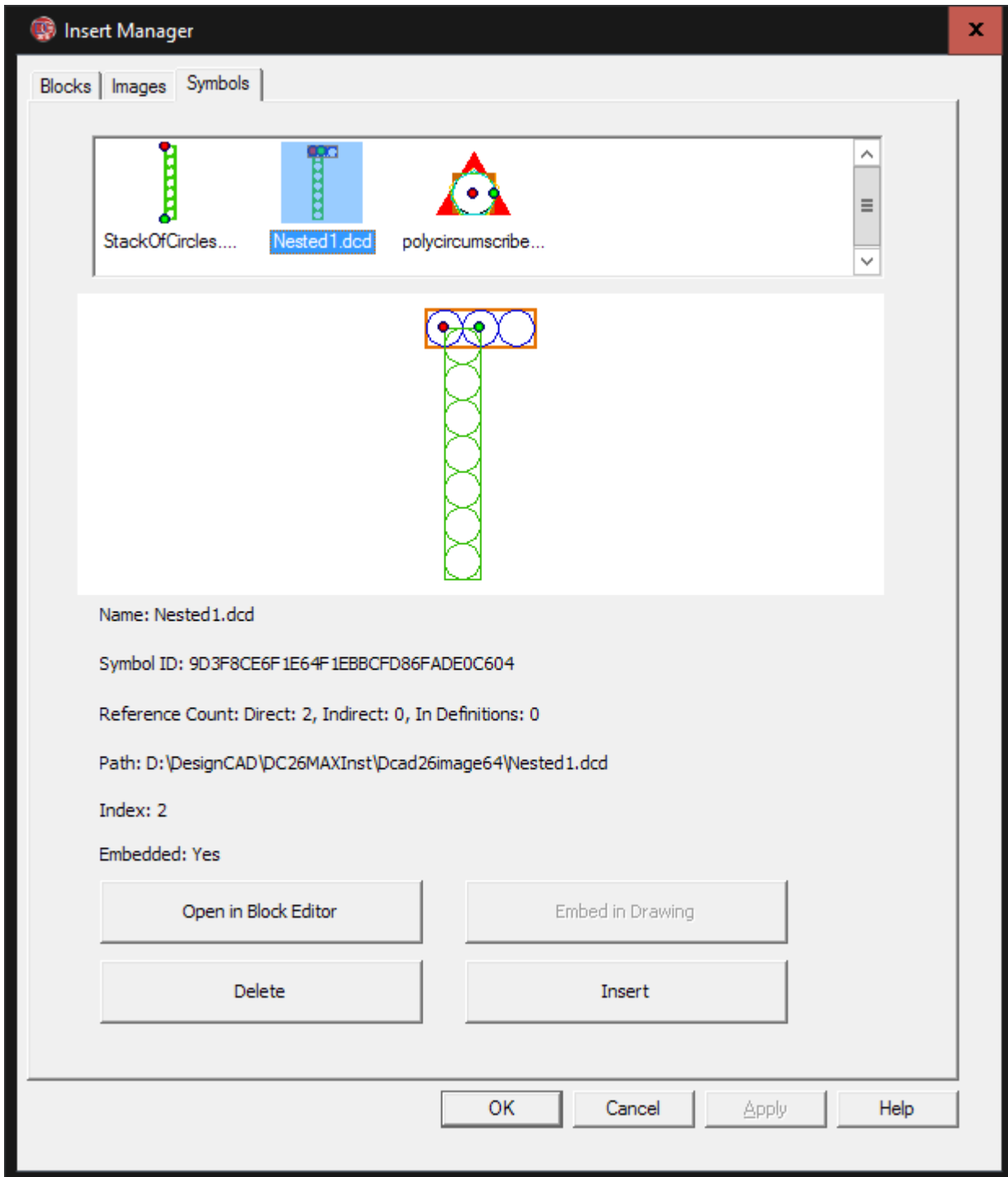
Als een ingesloten afbeelding is gemarkeerd in de visuele lijst, wordt de "**includen in de tekening**" knop gemarkeerd. Klikken op deze knop laat een referentie beeld ingebed worden in de tekening zodat het externe

beeldbestand niet meer nodig is. Dit verandert ook het **geïntegreerd** veld van **ja** naar **nee**. Let op, het oorspronkelijke beeldbestand moet aanwezig zijn in het opgegeven pad om dit te laten functioneren.



Symbolen (Symbols)

Het tabblad **Symbolen** van het invoeg beheer heeft dezelfde functies als het tabblad **Afbeeldingen** en heeft eveneens het **open in blok beheer** (Open in Block Editor) knop, net als het tabblad **Blokken** (Blocks).



Blok Beheer (Block Editor)

Wanneer u een symbool of blok markeert in het invoeg beheer ([Insert Manager](#)) en klikt op "Openen in blok beheer" (in Block Editor) dan wordt de inhoud van het blok of symbool weergegeven in een apart venster, zonder andere entiteiten in het zicht. Deze Blok beheer (Editor) modus heeft een ietwat beperkte set van teken- en bewerken commando's. Het is niet mogelijk tot het laden van symbolen of blokken via het menu Bestand, geen printopdracht, geen opties om het symbool of het blok als een extern bestand op te slaan en geen toegang tot speciale modi zoals papier kader ruimte modus (paper space modus) of Animatie en doorwandel modi. Er is ook geen toegang tot de hulp of het venster menu opdrachten en de opdracht opties hebben geen effect. Afgezien van deze uitzonderingen zijn de meeste andere tekenen en bewerken commando's beschikbaar voor u in het blok beheer.

Wilt u het invoegpunt van een blok of de handvatten van een symbool wijzigen in het Blok beheer, gebruik dan de **punt / toon teken handvatten** (Point / Show Drawing Handles) opdracht om de bestaande handgreep(en) van het blok of het symbool weer te geven. Gebruik de **Punt / instellen teken handvatten** (Point / Set Drawing Handles) opdracht om de bestaande hendel(s) naar de nieuwe locatie te verplaatsen.

Als u een symbool of blok wil toevoegen aan een die je wil bewerken, dan kan dit niet in het blok beheer bestands menu. In plaats daarvan selecteert u eerst het blok of symbool dat moet worden toegevoegd in het hoofdtekening venster en kopieer het naar het klembord. Dan kunt u het symbool of blok in het blok beheer openen en plak het in de extra content. **Opgelet, plak zeker geen kopie van een blok of symbool in zichzelf terwijl deze wordt aangepast - dit veroorzaakt een kringverwijzing en kan resulteren in een aantal lastige problemen.**

Om de wijzigingen in het blok beheer ongedaan te maken, kiest u **Bestand / Annuleer wijzigingen en sluit**. Om de wijzigingen in een blok of symbool in de Block Editor op te slaan kiest u **bestand / opslaan wijzigingen en sluit** (File / Save Changes and Exit).

Het invoeg beheer en het Blok beheer zijn beschikbaar in zowel model ruimte en papier kader ruimte. Er kunnen echter een aantal blokken en symbolen moeilijk te bewerken zijn in de "papier kader ruimte modus" vanwege het beperkte tekengebied dat beschikbaar is in deze modus. Als u problemen ondervindt bij het bewerken van een bepaald blok of een symbool in de "papier kader ruimte modus", probeer de "model ruimte" om het daar te bewerken.

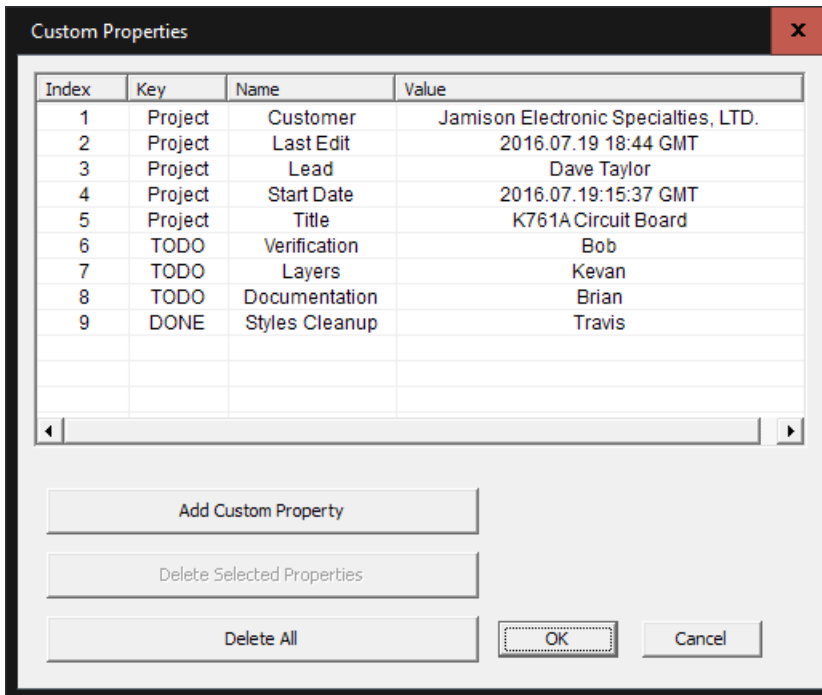
Aangepaste eigenschappen (Custom Properties)

DesignCAD ondersteunt nu het toevoegen van aangepaste eigenschappen aan tekeningen en teken entiteiten.

Aangepaste eigenschappen zijn stukjes tekst die worden gekenmerkt door een combinatie van een sleutel (Key) en een naam (Name) veld. Dit is een manier voor het toevoegen van "onzichtbare" tekstinformatie in een tekening of een object. Deze aangepaste eigenschappen kunnen worden bekeken en manueel bewerkt door middel van een aangepaste eigenschappen dialoog of verwerkt door code via BasicCAD macro's of OLE-automatisering-programma's.

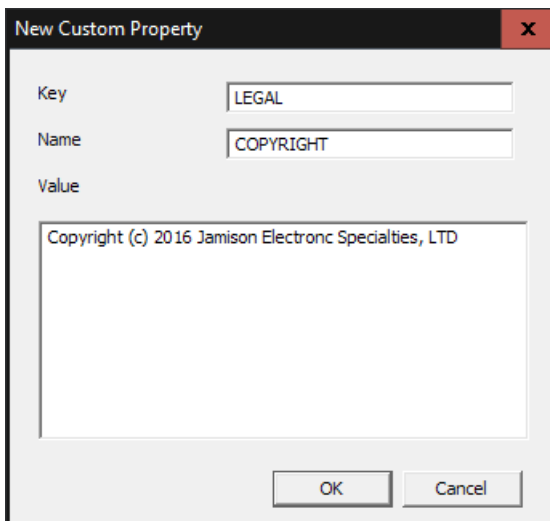
Aangepaste eigenschappen van de tekening zijn toegankelijk via het **bestand / document aangepaste eigenschappen** opdracht.

Aangepaste eigenschappen van de entiteit zijn toegankelijk via de knop **aangepaste eigenschappen** in de Info Box. Deze werkwijze opent een aangepaste eigenschappen editor:



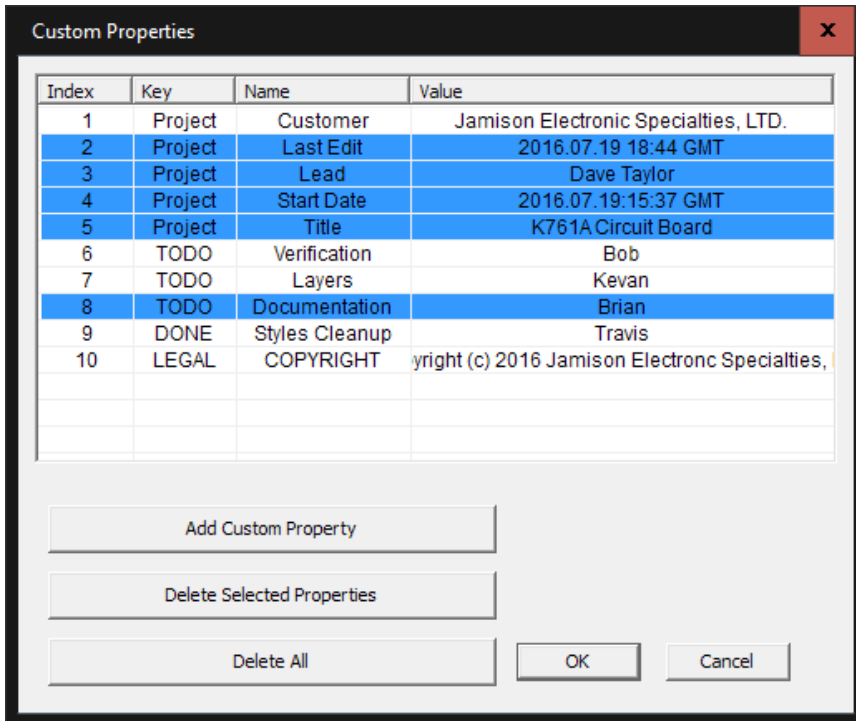
Het dialoogvenster toont vier velden: Index, Sleutel, Naam, en waarde (Index, Key, Name, and Value). De index is gewoon de volgorde waarin elke aangepaste eigenschap werd toegevoegd. De Sleutel (Key) is een van de twee velden die elke uniek aangepaste eigenschap identificeert. Het veld Naam is het tweede veld gebruikt om een aangepaste eigenschap te identificeren. Het waarde (Value) veld is de "inhoud" van de aangepaste eigenschap.

Een nieuwe aangepaste eigenschap (custom property) kan worden toegevoegd met behulp van de "**voeg toe aangepaste eigenschap**" knop (Add Custom Property).



Geen twee aangepaste eigenschappen voor een bepaalde tekening of tekening entiteit kan dezelfde Key / Naam combinatie hebben. Twee verschillende tekeningen kunnen identieke sleutel / naam (Key / Name) paren hebben en een tekening met een of meer entiteiten daarin opgenomen kunnen dezelfde sleutel / Naam paren hebben. De beperking geldt alleen voor de set van aangepaste eigenschappen in gebruik door een bepaalde tekening of een bijzondere tekening entiteit.

Een enkele aangepaste eigenschap kan worden geselecteerd door er links op te klikken. Meer kan worden geselecteerd met Ctrl-links-klik of door Shift-links-klik of Shift-pijl omlaag. Zodra een of meer aangepaste eigenschappen worden geselecteerd in het aangepaste eigenschappen beheer kunnen ze worden verwijderd met behulp van de **verwijder aangepaste eigenschappen** knop (Delete Selected Properties). De knop **verwijder alles** (Delete All) worden alle aangepaste eigenschappen die zijn toegevoegd via deze bewerking verwijderd.



Merk echter op dat sommige aangepaste eigenschappen die via BasicCAD of OLE-automatisering zijn toegevoegd niet bewerkbaar zijn. Een dergelijke niet-bewerkbare aangepaste eigenschap zal een grijze achtergrond hebben, en kan alleen worden gewijzigd of verwijderd met behulp van de BasicCAD of OLE-automatisering methoden.

Toegang tot aangepaste eigenschappen via BasicCAD en OLE-automatisering wordt hieronder beschreven.

Aangepaste eigenschappen van tekenobjecten worden gedupliceerd wanneer deze objecten worden gekopieerd en geplakt, gedupliceerd, gerangschikt, of getrimd. (Let op: alle niet-bewerkbare eigenschappen worden NIET gedupliceerd op deze manier.)

Wanneer u een verbonden teken object (compound drawing object) **explodeert** (zoals een dimensie of een rooster) dat aangepaste eigenschappen bevat, zullen de geëxplodeerde stukken niet meer deze eigenschappen bevatten. Dit is ook zo voor **gecombineerde Lijnen** (Combine Lines) en zorgt ervoor dat de resulterende objecten alle aangepaste eigenschappen verliezen die in zijn samenstellende delen kunnen voorkomen.

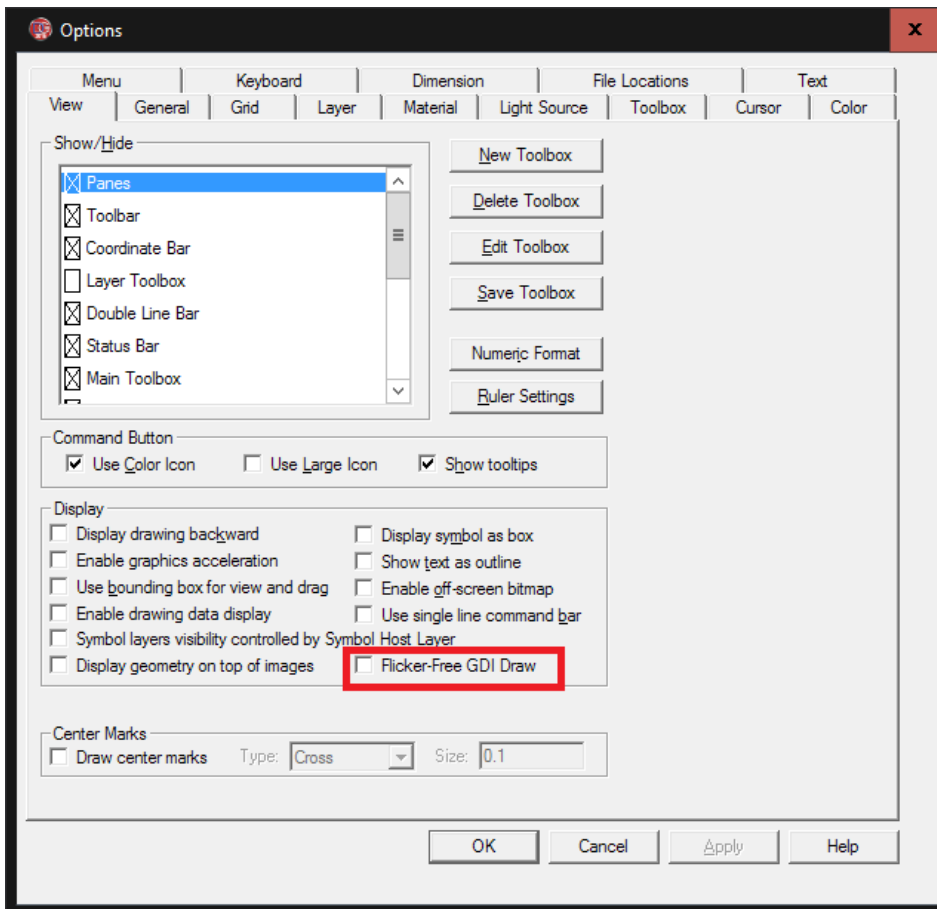
Aangepaste eigenschappen van tekeningen worden opgeslagen met de tekening wanneer u **Opslaan, Opslaan als, Een kopie opslaan, of Opslaan als symbool gebruikt**. Ze worden echter niet overgebracht in een gattkening indien u **laad symbool** gebruikt.

Andere wijzigingen in DesignCAD 2016

Flikkering-vrij GDI Teken

Een nieuwe optie is toegevoegd aan het weergave / opties (View Options) venster – Flikker vrij GDI Teken. Als deze optie is geselecteerd, maakt DesignCAD gebruik van een speciale optimalisering om het trekken van de grote lijnen te versnellen in het GDI draadframe uitzicht.

Deze optie heeft geen effect als de RedSDK mode actief is. Om toegang te krijgen tot deze instelling druk u op de "Q" toets om het opties paneel te openen en activeer het tabblad "Beeld". Het selectievakje is aan de onderkant rechts van de groep weergaveopties. Het verschil is bijzonder zichtbaar onder Windows 7 als u de doorzichtigheid (transparency) ingeschakeld hebt.



Algemene correcties en verbeteringen:

- Bij het exporteren van bestanden naar DWG of DXF-formaat, zou soms schuine tekst worden geconverteerd naar horizontale tekst. Dit is opgelost..
- De Scan Afbeelding opdracht was soms grijs omdat DesignCAD het benodigde programma niet kon vinden. Dit is opgelost..
- Bij de toepassing van een materiaal (zoals eiken, kersen, Parel, enz.) op een Solid oppervlakte entiteit, werd het materiaal niet opgeslagen met het object wanneer de tekening werd opgeslagen.
- Opslaan als 2D-projectie werkte niet met Solid oppervlak entiteiten.
- Werken met Multi-lijnen kon DesignCAD soms laten vastlopen. (crash)

BasicCAD Additions for Custom Properties

STATEMENT: ADDPROPERTY entid, key\$, name\$, value\$, nocopy

Arguments:

- **entid** – a number. If entid is greater than zero it represents the index of the entity to which the new property is to be added. If entid is 0, the property will be added to the drawing instead.
- **key\$** -- a string representing the key to be used in identifying this custom property.
- **name\$** – a string representing the name to be used in identifying this custom property.
- **value\$** – a string representing the contents of the property.
- **nocopy** – a flag that indicates whether this custom property should be propagated from the original entity via Duplicate, Copy/Paste, Array, or other editing operations. Also, if set, does not allow editing or deletion of this property via the Custom Property editor.

The ADDPROPERTY statement adds a new custom property to the specified entity if entid > 0, or to the drawing if entid = 0.

Errors:

- If the key/name pair clashes with an already-defined custom property, the statement triggers Error 114.
- If entid is non-zero and does not specify a legal entity id, the statement triggers Error 60.

Example 1:

```
' add a custom property to the first selected entity
PRECISION 0
ON ERROR GOTO oops
entid = GETSELECT 1, j
key$ = "PART"
name$ = "NUMBER"
value$ = "ZX_500_A12"
nocopy = 0 'false - this custom property can be edited by the drawing's user.
ADDPROPERTY entid, key$, name$, value$, nocopy
END

oops:
e = Error(q) ' get error number
MESSAGE "Error", e, "; ", Err$(e)
RETURN
```

Example 2:

```
' add a custom property to the drawing
entid = 0 ' specifies the drawing instead of an entity
key$ = "DRAWING"
name$ = "START DATE"
value$ = "06/25/2016"
nocopy = 0 'false - this custom property can be edited by the drawing's user.
ADDPROPERTY entid, key$, name$, value$, nocopy
```

STATEMENT: GETPROPERTY **entid**, **propid**, **key\$**, **name\$**, **value\$**, **nocopy**

Arguments:

- **entid** – a number. If entid is greater than zero it represents the index of the entity from which the property is to be retrieved. If entid is 0, the property will be retrieved from the drawing instead.
- **propid** – a number specifying the index of the property to be retrieved. Indexing begins at 1, not 0.
- **key\$** -- a string that will contain the key of this custom property.
- **name\$** – a string representing the name of this custom property.
- **value\$** – a string representing the contents of this custom property.
- **nocopy** – a flag that indicates whether this custom property can be propagated from the original entity via Duplicate, Copy/Paste, Array, or other editing operations. Also, if set, does not allow editing or deletion of this property via the Custom Property editor.

The GETPROPERTY statement retrieves a custom property from the specified entity if entid > 0, or from the drawing if entid = 0.

Errors:

- If entid is non-zero and does not specify a legal entity id, the statement triggers Error 60.
- If propid is not the index of an existing property, the statement triggers Error 25.

Example 1:

```
' get the third custom property from the first selected entity
e = 0
PRECISION 0
ON ERROR GOTO oops
entid = GETSELECT 1, j
propid = 3
key$ = "aaaa"
name$ = "bbbb"
value$ = "cccc"
nocopy = -1
GETPROPERTY entid, propid, key$, name$, value$, nocopy
IF e = 0 THEN
    MESSAGE "Property", propid, "    KEY:", key$, "; NAME:", name$, "; VALUE: ",
values$, "; nc: ", nocopy
END IF
END

oops:
e = Error(q) ' get error number
MESSAGE "Error", e, "; ", Err$(e)
RETURN
```

STATEMENT: SETPROPERTY **entid**, **propid**, **key\$**, **name\$**, **value\$**, **nocopy**

Arguments:

- **entid** – a number. If entid is greater than zero it represents the index of the entity for which the property is to be changed. If entid is 0, the indexed drawing property will be changed instead.
- **propid** – a number specifying the index of the property to be changed. Indexing begins at 1, not 0.
- **key\$** -- a string that will contain the new key of this custom property.
- **name\$** – a string representing the new name of this custom property.
- **value\$** – a string representing the new contents of this custom property.
- **nocopy** – a flag that indicates whether this custom property can be propagated from the original entity via Duplicate, Copy/Paste, Array, or other editing operations. Also, if set, does not allow editing or deletion of this property via the Custom Property editor.

The SETPROPERTY statement modifies a custom property from the specified entity if entid > 0, or from the drawing if entid = 0.

Errors:

- If entid is non-zero and does not specify a legal entity id, the statement triggers Error 60.
- If propid is not the index of an existing property, the statement triggers Error 25.
- If key\$ and name\$ would clash with an already-existing property, the statement triggers Error 114.

Example 1:

```
' Set the third custom property in the drawing
e = 0
PRECISION 0
ON ERROR GOTO oops
entid = 0 'we're targeting a drawing property instead of an entity property
propid = 3
key$ = "NewKey"
name$ = "NewName"
value$ = "I have changed."
nocopy = 1
SETPROPERTY entid, propid, key$, name$, value$, nocopy
IF e = 0 THEN
    GETPROPERTY entid, propid, nkey$, nname$, nvalue$, nnc
    MESSAGE "Updated Property", propid, " KEY:", nkey$, "; NAME:", nname$, _ ";
VALUE: ", nvalues$, "; nc: ", nnc
END IF
END

oops:
e = Error(q) ' get error number
MESSAGE "Error", e, "; ", Err$(e)
RETURN
```

STATEMENT: GETPROPERTYINDEX *entid*, *name\$*, *startindex*, *foundindex*

Arguments:

- **entid** – a number. If entid is greater than zero it represents the index of the entity to be searched. If entid is 0, the drawing properties will be searched instead.
- **name\$** – a string representing the custom property name to be searched for.
- **startindex** – a number representing which property index to start at.
- **foundindex** – a number representing the index of the next property that matches the specified name. Set to -1 if no matching property is found.

The GETPROPERTYINDEX statement searches the custom properties of the drawing (if entid is 0), or of the specified entity (if entid > 0) for a name field matching name\$. Set startindex to 1 to begin searching at the first custom property. If searching for multiple matches (which is possible, since there may be two or more properties with different key fields but the same name field), search for the second and subsequent matches by adding 1 to the last foundindex.

Errors:

- If entid is non-zero and does not specify a legal entity id, the statement triggers Error 60.
- If startindex is not the index of an existing property, the statement triggers Error 25.

Example 1:

```
' Search the drawing for all custom properties having the name "PARTNO"
entid = 0
name$ = "PARTNO"
startindex = 1
foundindex = 0
PRECISION 0
ON ERROR GOTO oops
DO WHILE foundindex >= 0
  GETPROPERTYINDEX entid, name$, startindex, foundindex
  IF e <> 0 THEN GOTO finish
  IF foundindex <> -1 Then
    MESSAGE "Match at index", foundindex
    Startindex = foundindex + 1
  END IF
LOOP
finish:
END

oops:
e = ERR(q)
RESUME NEXT
```

STATEMENT: DELETEPROPERTY *entid*, *propid*

Arguments:

- **entid** – a number. If entid is greater than zero it represents the index of the entity from which a custom property is to be deleted. If entid is 0, a custom property will be deleted from the drawing properties instead.
- **propid** – the index of the custom property to be deleted.

The DELETEPROPERTY statement deletes the custom property specified by propid from the entity specified by entid, or from the drawing if entid is zero.

Errors:

- If entid is non-zero and does not specify a legal entity id, the statement triggers Error 60.
- If propid is not the index of an existing property, the statement triggers Error 25.

Example 1:

```
' Delete the fourth custom property from entity 2
entid = 2
propid = 4
DELETEPROPERTY entid, propid
```

New Sys() functions relating to Custom Properties

Sys(689) – the number of custom properties that the current drawing contains.

Sys(690) – the number of custom properties contained in the entity last loaded by the ENTITY statement.

CustomProperties Class

Methods:

AddProperty(Key\$, Name\$, Value\$, bNoCopy) as Integer

Returns the index of the newly-added property if successful, or -1 in case of failure.

DeleteProperty(iIndex) as Boolean

Returns True if successful or False if the property could not be deleted for any reason.

Item(iIndex) as CustomProperty

Returns the CustomProperty specified by the index.

Properties:

Count as Long

Returns the number of custom properties in the collection

CustomProperty Class

Methods:

SetKey(Key\$) as Boolean

Sets the Custom Property's Key to the value specified in Key\$. Returns False if the key could not be updated to the specified value (typically due to a key/name clash that would result).

SetName(Name\$) as Boolean

Sets the Custom Property's Name to the value specified in Name\$. Returns False if the Name could not be updated to the specified value (typically due to a key/name clash that would result).

Properties:

Key as String (read-only)

Returns the Key for the Custom Property. Read-only. To change the contents of the Key property, use the SetKey method.

Name as String (read-only)

Returns the Name for the Custom Property. Read-only. To change the contents of the Name property, use the SetName method.

Data as String (read/write)

Returns the Data contents for the Custom Property. Can be directly read and written.

NoCopy as Boolean

Returns the NoCopy field of the Custom Property. Can be set directly read and written.