# Chapter 8
## Functions that Work all by Themselves!

**Samples In this Chapter Include:**

# Chapter 8 Single Action Functions

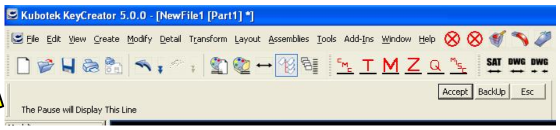
*Single action functions* are functions that don't need other functions to work. Although these functions are single action there really needs to be another function that will allow you to see the function work.
In the samples in this chapter we will show you the sample and the result of the function. What that means is even though it is a single action function there really needs to be a second function that will show you the result. When you go to the next chapter you will not need to see the result you can just move on to the next function in your code.

We will be using the Pause function for most of these results, we will start with that function first. If you read the syntax for the Pause function then you will have a good idea about where we are heading.

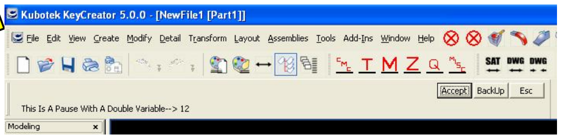
## Pause

The Pause command will "pause" the program and display a prompt on the menu line. That prompt can be a predefined prompt or a string developed from variables in the command line.

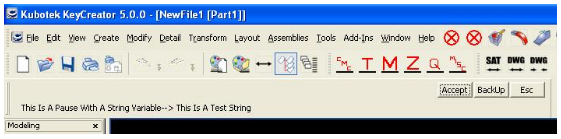
```
Double Test
String $strTest

Test = 12
$strTest = "This Is A Test String"

//First Pause Line
Pause "The Pause will Display This Line"

//Second Pause Line
Pause "This Is A Pause With A Double Variable--> %d",Test

//Third Pause Line
Pause "This Is A Pause With A String Variable--> %s",$strTest

Exit
```


Kubotek KeyCreator 5.0.0 - [NewFile1 [Part1] *]
File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
Accept BackUp Esc

```
The Pause will Display This Line
```


Kubotek KeyCreator 5.0.0 - [NewFile1 [Part1]]
File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
Accept BackUp Esc
Modeling

```
This Is A Pause With A Double Variable--> 12
```


Kubotek KeyCreator 5.0.0 - [NewFile1 [Part1]]
File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
Accept BackUp Esc
Modeling

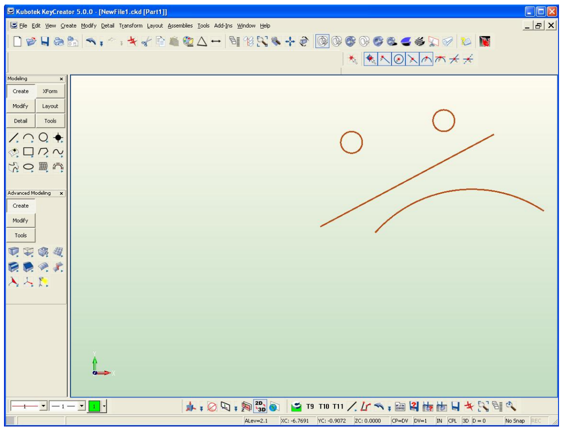
```
This Is A Pause With A String Variable--> This Is A Test String
```

## Auto

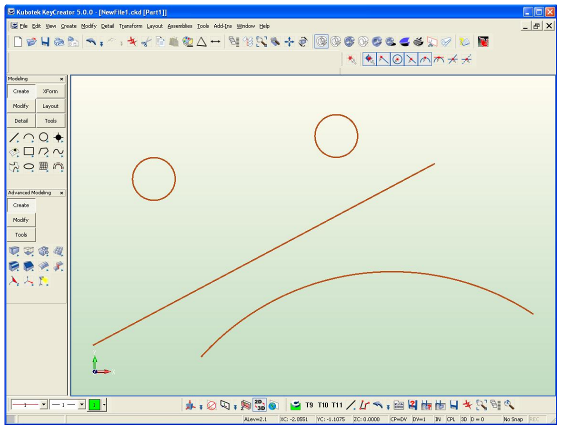
The *Auto* function will simply auto size the view port to all geometry and text in the drawing that is displayed in the level list. This is helpful if you want to print all then you can *auto* scale the view port and then print.

```
Int nVP
SET DISPVIEW, 1, nVP

AUTO

EXIT
```

Before



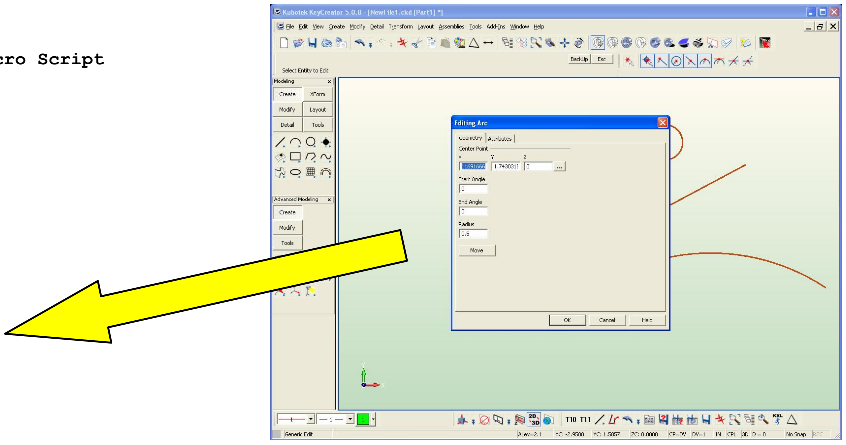
After



## CallFunc

The CallFunc function gives you the ability to call KeyCreator commands from in side of your KXL program. One thing to remember is that not all KeyCreator commands work well with KXL programs, the problem is that there are two types of KC commands non-immediate mode and immediate mode.
If the function is an immediate function, the KXL will continue after the function finishes unless the user escapes.

```
// Begin KXL Record
// This kxl is intended to be a Macro Script

MASKCLEAR

// (0xAA25) Edit Entities
NoteState
MASKCLEAR

// End KXL Record



MASKCLEAR
CallFunc (0xAA25) //Edit Entities
NoteState
MASKCLEAR
Exit
```



There is currently no list of commands that work for the CallFunc function so the best way to "find" them is to record a macro with the function that you need and then open the KXL file and look at the code for the CallFunc command line there you will see the number of the command.

## Clear

The clear function must be used with caution. If used by itself then this function will clear all variables at the time of the execution. If you only want to clear one or two variables then you will need to use the clear command with the variable after the command.

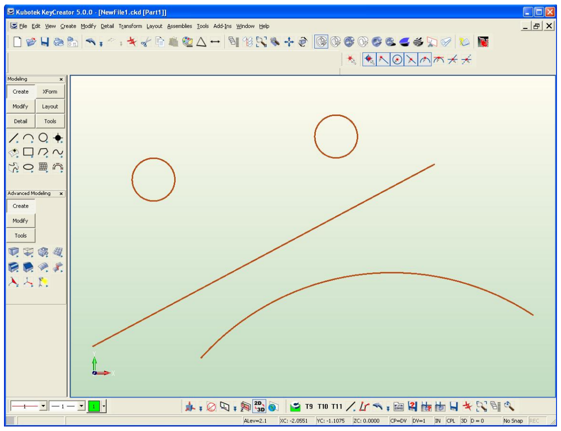
```
Double Test
String $strTest

Test = 12
$strTest = "This Is A Test String"

//First Pause Line
Pause "The Pause will Display This Line"

//Second Pause Line
Pause "This Is A Pause With A Double Variable--> %d",Test

Clear
// This Will cause an error since this will empty the variable $strTest

//Third Pause Line
Pause "This Is A Pause With A String Variable--> %s",$strTest

Exit
```

## DblScale

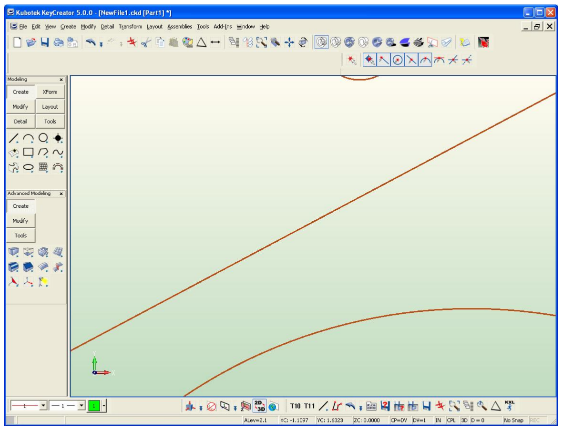
The dblscale function is designed to zoom in double the view port size. With the parameters filled you can manipulate just the active view port or all view ports.

```
Int nVP
SET DISPVIEW, 1, nVP

AUTO

DBLSCL

EXIT
```

Before



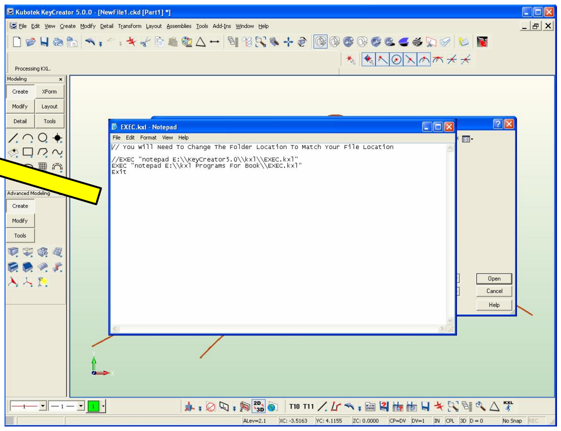
After



## Exec

The Exec statement allows external programs or system processes to be accessed and run from inside the KXL program. When executed, this command temporarily suspends the system and executes the requested command. This function can be used with any computer program. Such as the example it will open the internet and go to a page for looking up sheet metal gages then return you back to the KXL program.

```
// You Will Need To Change The Folder Location To
// Match Your File Location

EXEC "notepad C:\\KeyCreator5.0\\kxl\\EXEC.kxl"

Exit
```



```
// You Will Need To Change The Folder Location To Match Your File Location

//EXEC "notepad E:\\KeyCreator5.0\\kxl\\EXEC.kxl"
EXEC "notepad E:\\kxl Programs For Book\\EXEC.kxl"
Exit
```

## Exit

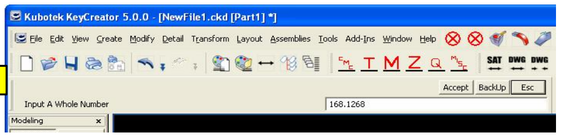
Exit causes an immediate exit from the KXL file being executed. This is a basic function that can be used in conjunction with conditional statements to exit out of the KXL program.

```
Double Test
String $strTest

Test = 12
$strTest = "This Is A Test String"

//First Pause Line
Pause "The Pause will Display This Line"

// This Will Exit The KXL Instantly
Exit

//Second Pause Line
Pause "This Is A Pause With A Double Variable--> %d",Test

//Third Pause Line
Pause "This Is A Pause With A String Variable--> %s",$strTest

Exit
```

## GetDBL

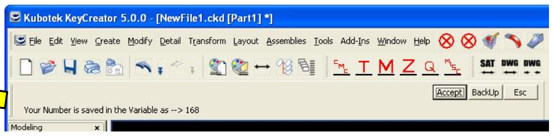
The GetDBL function will prompt the user to input a number. Only a number can be input in the field an error will result if you place a letter of anything other than a number. The inputted number will be placed in the variable assigned to the getdbl function line.

```
Double Test
Test = 12

GETdbl "Input A Whole Number",Test,Test
```


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1] *]
File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
Accept BackUp Esc
Modeling

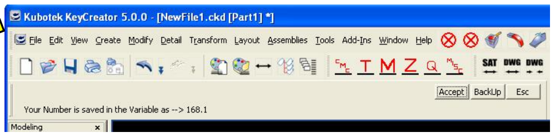
Input A Whole Number 168.1268

```
// This will Show you the Whole Number
Pause "Your Number is saved in the Variable as --> %d",Test
```


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1] *]
File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
Accept BackUp Esc
Modeling

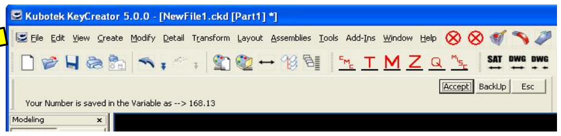
Your Number is saved in the Variable as --> 168

```
// This will Show you the Whole Number + 1 decimal
Pause "Your Number is saved in the Variable as --> %.1f",Test
```


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1] *]
File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
Accept BackUp Esc
Modeling

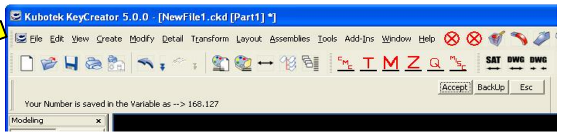
Your Number is saved in the Variable as --> 168.1

```
// This will Show you the Whole Number + 2 decimal
Pause "Your Number is saved in the Variable as --> %.2f",Test
```


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1] *]
File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
Accept BackUp Esc
Modeling

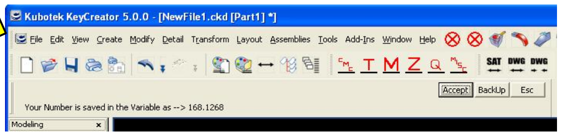
Your Number is saved in the Variable as --> 168.13

```
// This will Show you the Whole Number + 3 decimal
Pause "Your Number is saved in the Variable as --> %.3f",Test
```


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1] *]
File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
Accept BackUp Esc
Modeling

Your Number is saved in the Variable as --> 168.127

```
// This will Show you the Whole Number + 4 decimal
Pause "Your Number is saved in the Variable as --> %.4f",Test

Exit
```


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1] *]
File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
Accept BackUp Esc
Modeling

Your Number is saved in the Variable as --> 168.1268

## GetDisplayView

The GetDisplayView function retrieves the specified system view matrix and stores it in the data array or the system array @dbldat. Use this to get the matrix of a predefined view.

```
Double vData[0]
Array vData[15]

GetDispView 0, vData

Pause"vData[0] Is-->%.5f" ,vData[0]
Pause"vData[1] Is-->%.5f" ,vData[1]
Pause"vData[2] Is-->%.5f" ,vData[2]
Pause"vData[3] Is-->%.5f" ,vData[3]
Pause"vData[4] Is-->%.5f" ,vData[4]
Pause"vData[5] Is-->%.5f" ,vData[5]
Pause"vData[6] Is-->%.5f" ,vData[6]
Pause"vData[7] Is-->%.5f" ,vData[7]
Pause"vData[8] Is-->%.5f" ,vData[8]
Pause"vData[9] Is-->%.5f" ,vData[9]
Pause"vData[10] Is-->%.5f" ,vData[10]
Pause"vData[11] Is-->%.5f" ,vData[11]
Pause"vData[12] Is-->%.5f" ,vData[12]
Pause"vData[13] Is-->%.5f" ,vData[13]
Pause"vData[14] Is-->%.5f" ,vData[14]
Pause"vData[15] Is-->%.5f" ,vData[15]

Exit
```

*Note:*
*I have paused each line to have it display to you each matrix variable. In actual KXL code you will not need the Pause lines or to display the variables.*

## GetLevel

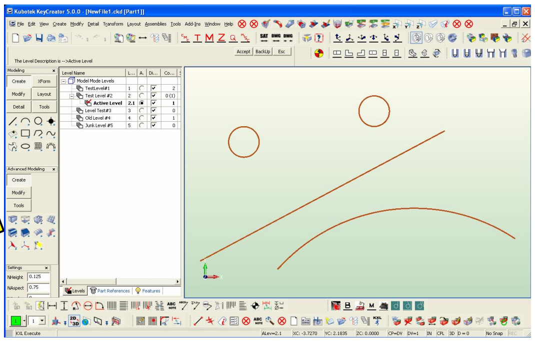
The GetLevel will return a handle to the currently active level. However this function is useless by its self so you will see in the next sample that we are using the get level function just to get the handle for the next called function.

```
CLEAR hLev
HLEVEL hLev

GETLEVEL  hLev

Exit
```

## GetLevelInfo

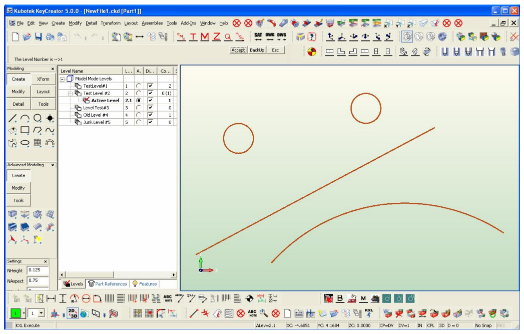
The GetLevelInfo uses the previous function getLevel. This function returns all information pertaining to a Level.

```
CLEAR hLev
HLEVEL hLev
String $strDescriptor, $strLevelNumberString
Int nLevelnumber

GETLEVEL  hLev

GETLEVELINFO hLev, $strDescriptor, $strLevelNumberString,
nLevelnumber, hParentLevel

Pause "The Level Description is -->%s",$strDescriptor
Pause "The Level Number String is -->%s",$strLevelNumberString
Pause "The Level Number is -->%i",nLevelnumber


Exit
```


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1]]

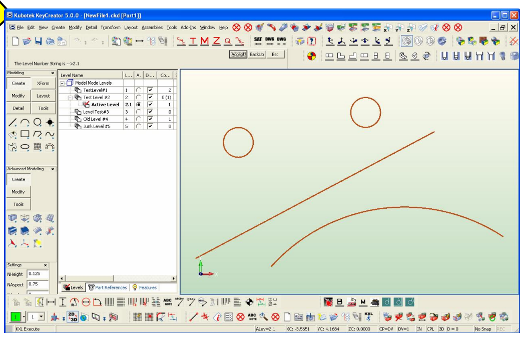
The Level Description is -->Active Level

| Level Name | L... | A. | Di... | Co... |
|---|---|---|---|---|
| Model Mode Levels | | | | |
| TestLevel#1 | 1 | ○ | ✓ | 2 |
| Test Level #2 | 2 | ○ | ✓ | 0 (1) |
| **Active Level** | **2.1** | ● | ✓ | **1** |
| Level Test#3 | 3 | ○ | ✓ | 0 |
| Old Level #4 | 4 | ○ | ✓ | 1 |
| Junk Level #5 | 5 | ○ | ✓ | 0 |


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1]]

The Level Number is -->1

| Level Name | L... | A. | Di... | Co... |
|---|---|---|---|---|
| Model Mode Levels | | | | |
| TestLevel#1 | 1 | ○ | ✓ | 2 |
| Test Level #2 | 2 | ○ | ✓ | 0 (1) |
| **Active Level** | **2.1** | ● | ✓ | **1** |
| Level Test#3 | 3 | ○ | ✓ | 0 |
| Old Level #4 | 4 | ○ | ✓ | 1 |
| Junk Level #5 | 5 | ○ | ✓ | 0 |


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1]]

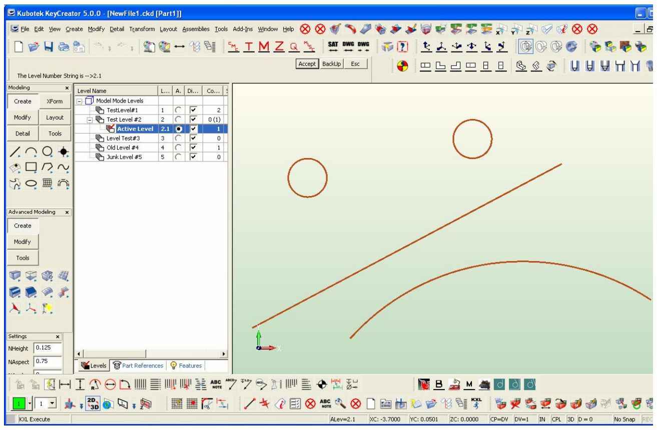
The Level Number String is -->2.1

| Level Name | L... | A. | Di... | Co... |
|---|---|---|---|---|
| Model Mode Levels | | | | |
| TestLevel#1 | 1 | ○ | ✓ | 2 |
| Test Level #2 | 2 | ○ | ✓ | 0 (1) |
| **Active Level** | **2.1** | ● | ✓ | **1** |
| Level Test#3 | 3 | ○ | ✓ | 0 |
| Old Level #4 | 4 | ○ | ✓ | 1 |
| Junk Level #5 | 5 | ○ | ✓ | 0 |

## GetLevelNumber

The GetLevelNumber is just like the GetLevelInfo Function only that it will only retrieve the level number string. If the level is on a sub level it will not retrieve the parent level but the complete level string.

```
CLEAR hLev
HLEVEL hLev
String $strLevNum

GETLEVEL  hLev

GetLevelNumber hLev, $strLevNum

Pause "The Level Number String is -->%s",$strLevNum

Exit
```

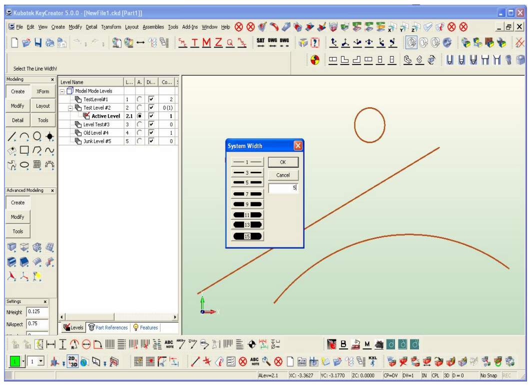

Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1]]

The Level Number String is -->2.1

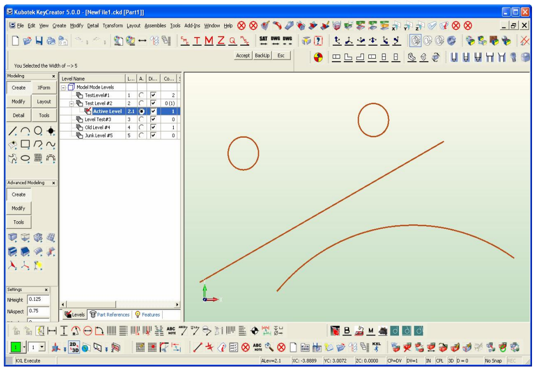
## GetLWidth

The GetLWidth displays a text string on the prompt line along with a window of line width icons and waits for a line width to be selected. You can use this as a way to graphically let the user select the line width. Use this in conjunction with the SetLWidth.

```
Int nRetType

GetLWidth "Select The Line Width!", nRetType

Pause "You Selected the Width of --> %i",nRetType

Exit
```


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1]]

Select The Line Width!

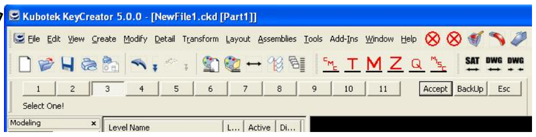

Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1]]

You Selected the Width of --> 5

## GetMenu

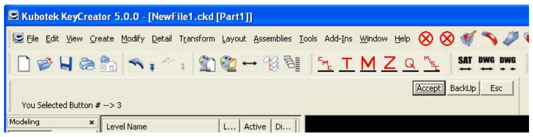
The GetMenu displays a text string on the prompt line and a menu in the menu area, then waits until a menu item is selected. The menu item chosen is returned in the system variable @KEY.

```
GetMenu "Select One!","1|2|3|4|5|6|7|8|9|10|11",3

Pause "You Selected Button # --> %i",@key

Exit
```


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1]]

Select One!

File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
1 2 3 4 5 6 7 8 9 10 11
Accept BackUp Esc
Modeling
Level Name L... Active Di...


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1]]

You Selected Button # --> 3

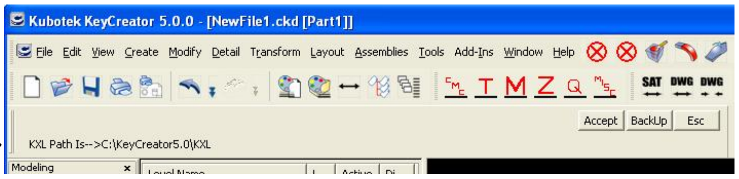
File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
Accept BackUp Esc
Modeling
Level Name L... Active Di...

The number in the third variable is for the default menu item. You can set this to any menu position you need to point to the preselected menu item.

## GetPath

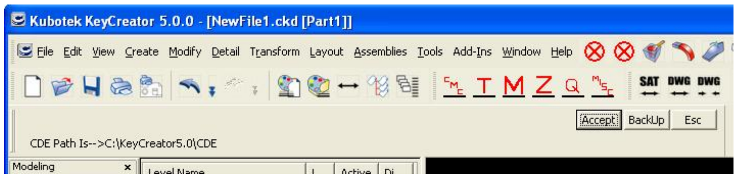
Use the GetPath to retrieve the current value of a registered path. You can use this or each path that you want to get or multiples, you will just have to store each path in its own variable.

```
Clear
String $Path1, $Path2, $Path3, $Path4
String $Path5, $Path6, $Path7, $Path
String $Path9, $Path10, $Path11, $Path12
String $Path13, $Path14, $Path15, $Path16
String $Path17

GetPath "KXL",$Path1
GetPath "CDE",$Path2
GetPath "DWG",$Path3
GetPath "DXF",$Path4
GetPath "DesignFiles",$Path5
GetPath "IGES",$Path6
GetPath "Notes",$Path7
GetPath "Parasolid",$Path8
GetPath "SAT",$Path9
GetPath "STEP",$Path10
GetPath "STL",$Path11
GetPath "Templates",$Path12
GetPath "Textures",$Path13
GetPath "VRML",$Path14
GetPath "CATIA",$Path15
GetPath "ProE",$Path16
GetPath "UG",$Path17

Pause"KXL Path Is-->%s" ,$Path1
Pause"CDE Path Is-->%s" ,$Path2
Pause"DWG Path Is-->%s" ,$Path3
Pause"DXF Path Is-->%s" ,$Path4
Pause"DesignFiles Path Is-->%s" ,$Path5
Pause"IGES Path Is-->%s" ,$Path6
Pause"Notes Path Is-->%s" ,$Path7
Pause"Parasolid Path Is-->%s" ,$Path8
Pause"SAT Path Is-->%s" ,$Path9
Pause"STEP Path Is-->%s" ,$Path10
Pause"STL Path Is-->%s" ,$Path11
Pause"Templates Path Is-->%s" ,$Path12
Pause"Textures Path Is-->%s" ,$Path13
Pause"VRML Path Is-->%s" ,$Path14
Pause"CATIA Path Is-->%s" ,$Path15
Pause"ProE Path Is-->%s" ,$Path16
Pause"UG Path Is-->%s" ,$Path17

Exit
```


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1]]

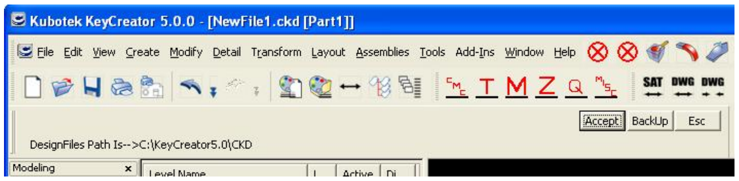
KXL Path Is-->C:\KeyCreator5.0\KXL

File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
Accept BackUp Esc
Modeling


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1]]

CDE Path Is-->C:\KeyCreator5.0\CDE

File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
Accept BackUp Esc
Modeling


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1]]

DesignFiles Path Is-->C:\KeyCreator5.0\CKD

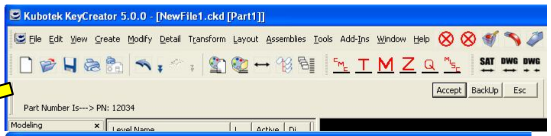
File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
Accept BackUp Esc
Modeling

I have only showed you three screen shots however this sample will display all paths in the tools > options > directories page.

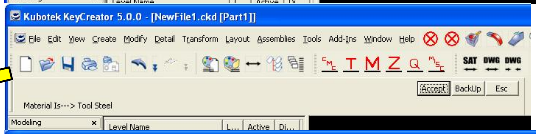
## GetSTDPropFile

This function gets the standard properties for the active design FILE. This function works nicely for creating bill of materials or even placing notes in your drawings. In order for this sample to work properly you will have to fill in the fields in the standard part properties dialog box.

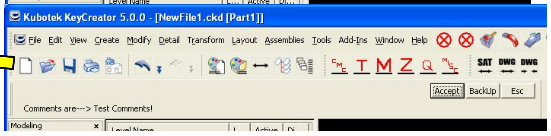
```
String $PartNumber, $Description, $Creator
String $Keywords, $LastAuthor
String $Revision, $Material, $Vendor, $Comments
String $Created, $LastSaved

GetSTDPropFile $PartNumber, $Description, $Creator,
$Keywords, $LastAuthor, $Revision, $Material, $Vendor,
$Comments, $Created, $LastSaved

Pause "Part Number Is---> %s", $PartNumber
Pause "Description Is---> %s", $Description
Pause "Creator Is---> %s", $Creator
Pause "Keywords are---> %s", $Keywords
Pause "Last Author Is---> %s", $LastAuthor
Pause "Revision Is---> %s", $Revision
Pause "Material Is---> %s", $Material
Pause "Vendor Is---> %s", $Vendor
Pause "Comments are---> %s", $Comments
Pause "Created By---> %s", $Created
Pause "Last Saved By ---> %s", $LastSaved

Exit
```



Part Number Is---> PN: 12034



Material Is---> Tool Steel



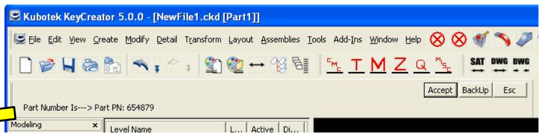
Comments are---> Test Comments!

Once again I have only shown three screen shots however you can get just one field or multiples. If you are only going to get one field value then you will need to place the proper number of commas in the command line up to the field variable you are wanting.

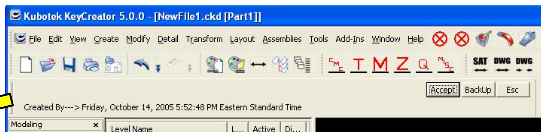
## GetSTDPropPart

This function gets the standard properties for the active design PART. This function works nicely for creating bill of materials or even placing notes in your drawings. In order for this sample to work properly you will have to fill in the fields in the standard part properties dialog box.

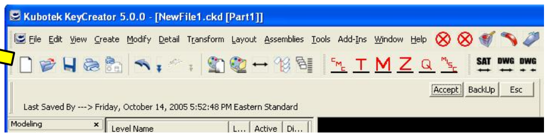
```
String $PartNumber, $Description, $Creator
String $Keywords, $LastAuthor
String $Revision, $Material, $Vendor, $Comments
String $Created, $LastSaved

GetSTDPropFile $PartNumber, $Description, $Creator,
$Keywords, $LastAuthor, $Revision, $Material, $Vendor,
$Comments, $Created, $LastSaved

Pause "Part Number Is---> %s", $PartNumber
Pause "Description Is---> %s", $Description
Pause "Creator Is---> %s", $Creator
Pause "Keywords are---> %s", $Keywords
Pause "Last Author Is---> %s", $LastAuthor
Pause "Revision Is---> %s", $Revision
Pause "Material Is---> %s", $Material
Pause "Vendor Is---> %s", $Vendor
Pause "Comments are---> %s", $Comments
Pause "Created By---> %s", $Created
Pause "Last Saved By ---> %s", $LastSaved

Exit
```



Part Number Is---> Part PN: 654879



Created By---> Friday, October 14, 2005 5:52:48 PM Eastern Standard Time



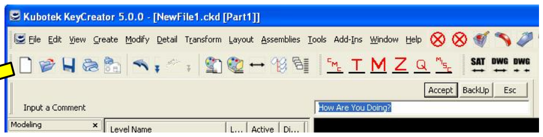
Last Saved By ---> Friday, October 14, 2005 5:52:48 PM Eastern Standard

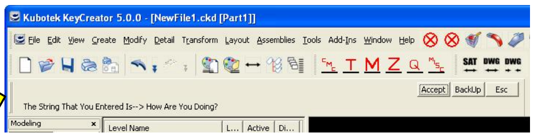
This is just like the GetstdPropFile function I have only shown three screen shots however you can get just one field or multiples. If you are only going to get one field value then you will need to place the proper number of commas in the command line up to the field variable you are wanting.

## GetSTR

This function is just like the GetDBL function except it is of Alfa-numeric values. The GetStr displays a text string on the prompt line and a menu in the menu area then waits for either a menu item to be selected or a text string to be entered. If a menu item is selected, the number is returned in the system variable @KEY. If a text string is entered instead, its value is stored in the specified string variable.

```
String $strCom
$strCom = "Hello, There!"

GetSTR "Input a Comment",$strCom,$strCom

Pause "The String That You Entered Is--> %s",$strCom

Exit
```

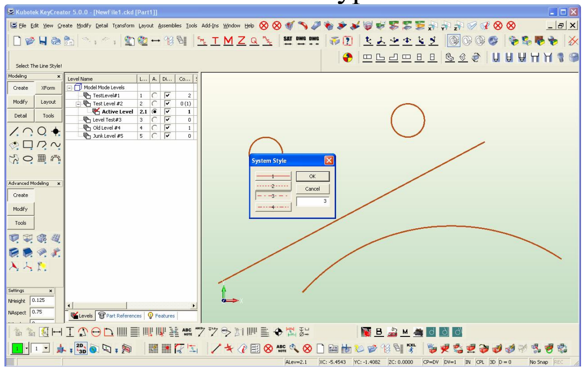

Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1]]
Input a Comment
How Are You Doing?
File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
Accept BackUp Esc
Modeling
Level Name L... Active Di...

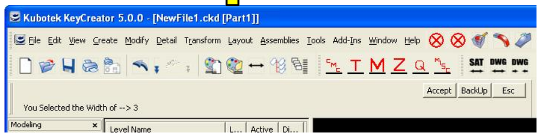

Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1]]
The String That You Entered Is--> How Are You Doing?
File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
Accept BackUp Esc
Modeling
Level Name L... Active Di...

## GetLStyle

The GetLStyle is like the GetLWidth function. This function displays a text string on the prompt line along with a window of line style icons and waits for one or more line types to be selected.

```
Int nRetType

GetLStyle "Select The Line Style!", nRetType

Pause "You Selected the Width of --> %i",nRetType

Exit
```

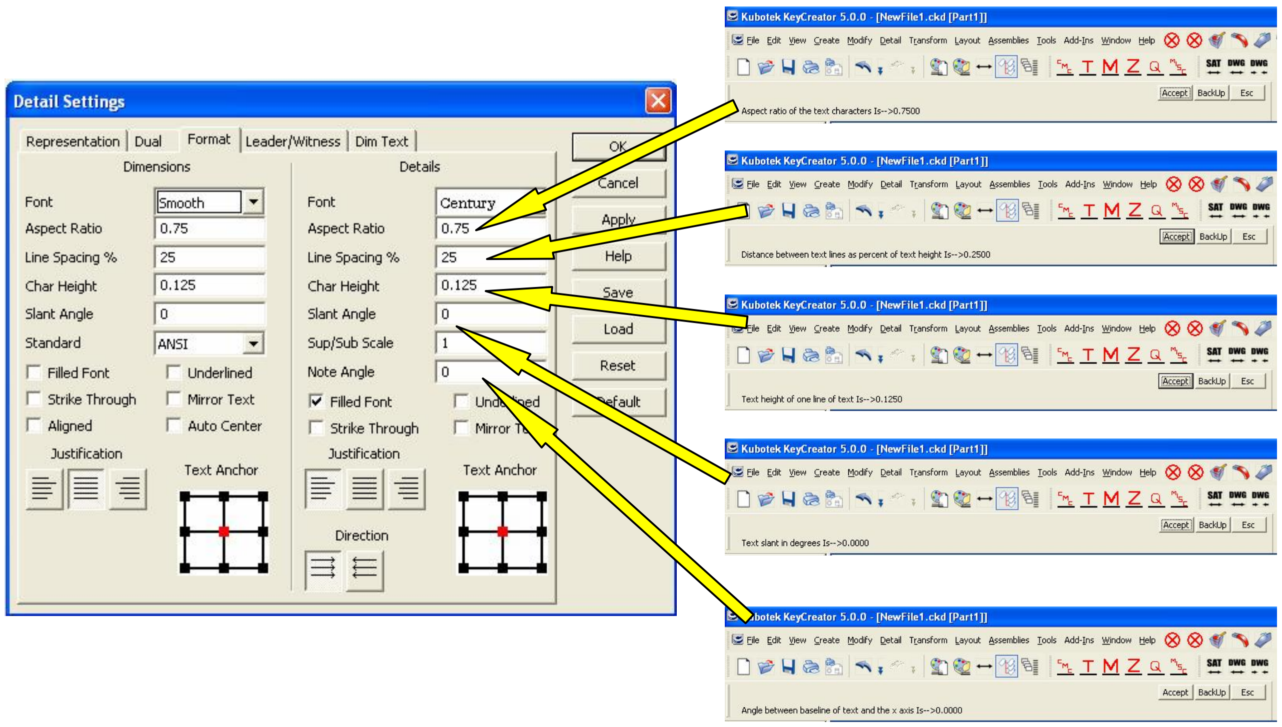

Select The Line Style!


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1]]
You Selected the Width of --> 3
File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
Accept BackUp Esc
Modeling
Level Name L... Active Di...

## GetTextAttribs

This function will return the active system text attributes used for any newly created text.

```
Double adTxtAtt[0]
Array adTxtAtt[13]

GetTextAttribs adTxtAtt

Pause "Angle between baseline of text and the x axis Is-->%.4f", adTxtAtt[0]
Pause "Text height of one line of text Is-->%.4f", adTxtAtt[1]
Pause "Aspect ratio of the text characters Is-->%.4f", adTxtAtt[2]
Pause "Distance between text lines as percent of text height Is-->%.4f", adTxtAtt[3]
Pause "Text slant in degrees Is-->%.4f", adTxtAtt[4]
Pause "Alignment of the text Is-->%d", adTxtAtt[5]
Pause "Text direction: Is-->%d", adTxtAtt[6]
Pause "Horizontal alignment Is-->%d", adTxtAtt[7]
Pause "Vertical alignment Is-->%d", adTxtAtt[8]
Pause "The text is mirrored Is-->%d", adTxtAtt[9]
Pause "Filled text Is-->%d", adTxtAtt[10]
Pause "The text is underlined Is-->%d", adTxtAtt[11]
Pause "The text is strike through Is-->%d", adTxtAtt[12]
Exit
```



Detail Settings

Representation | Dual | Format | Leader/Witness | Dim Text

| Dimensions | | Details | |
|---|---|---|---|
| Font | Smooth | Font | Century |
| Aspect Ratio | 0.75 | Aspect Ratio | 0.75 |
| Line Spacing % | 25 | Line Spacing % | 25 |
| Char Height | 0.125 | Char Height | 0.125 |
| Slant Angle | 0 | Slant Angle | 0 |
| Standard | ANSI | Sup/Sub Scale | 1 |
| | | Note Angle | 0 |

Dimensions: ☐ Filled Font ☐ Underlined ☐ Strike Through ☐ Mirror Text ☐ Aligned ☐ Auto Center

Details: ☑ Filled Font ☐ Underlined ☐ Strike Through ☐ Mirror Text

Justification
Text Anchor
Direction
OK
Cancel
Apply
Help
Save
Load
Reset
Default
Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1]]
File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
Accept BackUp Esc

Aspect ratio of the text characters Is-->0.7500

Distance between text lines as percent of text height Is-->0.2500

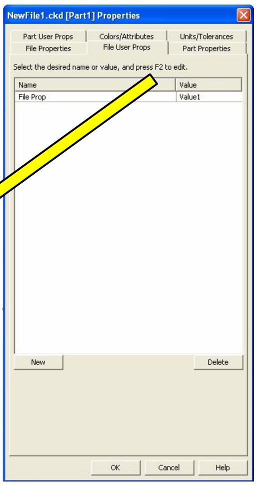
Text height of one line of text Is-->0.1250

Text slant in degrees Is-->0.0000

Angle between baseline of text and the x axis Is-->0.0000

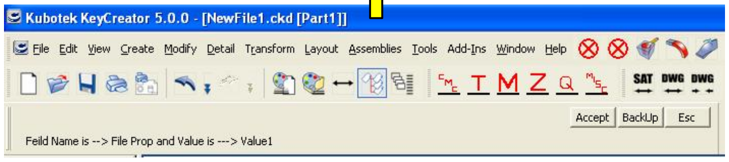
## GetUserPropFile

This function will "get" the user properties for the active design file. The function will place the field name and the corresponding value in the same array number. Example, the first field name will be in `$$aNames[0]` and the first value will be in `$$aValues[0]`. Caution should be used since if the value for the name field is empty then it is possible that the first value may actually become the second value.



| Name | Value |
|---|---|
| File Prop | Value1 |

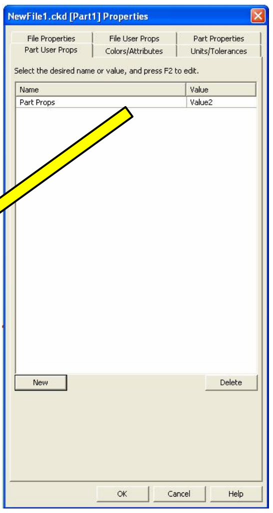
```
SList $$aNames,$$aValues

GetUserPropFile $$aNames, $$aValues

Pause "Feild Name is --> %s and Value is ---> %s", $$aNames[0], $$aValues[0]

Exit
```


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1]]
File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
SAT DWG DWG
Accept BackUp Esc

Feild Name is --> File Prop and Value is ---> Value1

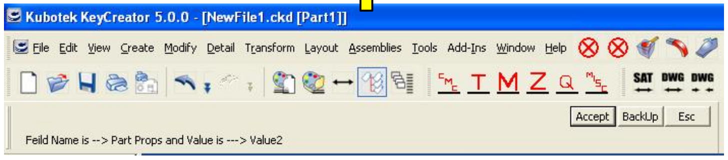
## GetUserPropPart

This function will "get" the user properties for the active design file. The function will place the field name and the corresponding value in the same array number. Example, the first field name will be in `$$aNames[0]`and the first value will be in `$$aValues[0]`. Caution should be used since if the value for the name field is empty, then it is possible that the first value may actually become the second value.



| Name | Value |
|---|---|
| Part Props | Value2 |

```
SList $$aNames,$$aValues

GetUserPropPart $$aNames, $$aValues

Pause "Feild Name is --> %s and Value is ---> %s", $$aNames[0],
$$aValues[0]

Exit
```


Kubotek KeyCreator 5.0.0 - [NewFile1.ckd [Part1]]
File Edit View Create Modify Detail Transform Layout Assemblies Tools Add-Ins Window Help
SAT DWG DWG
Accept BackUp Esc

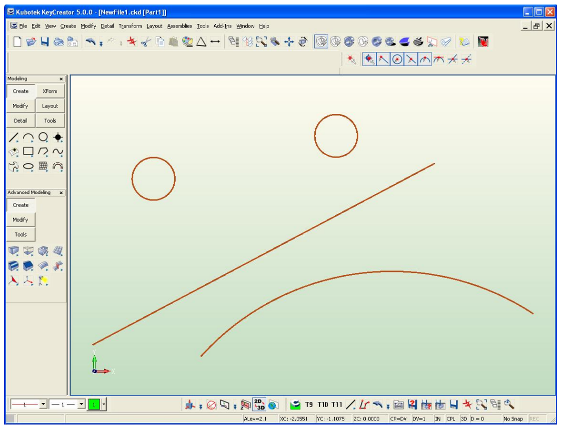
Feild Name is --> Part Props and Value is ---> Value2

## Half

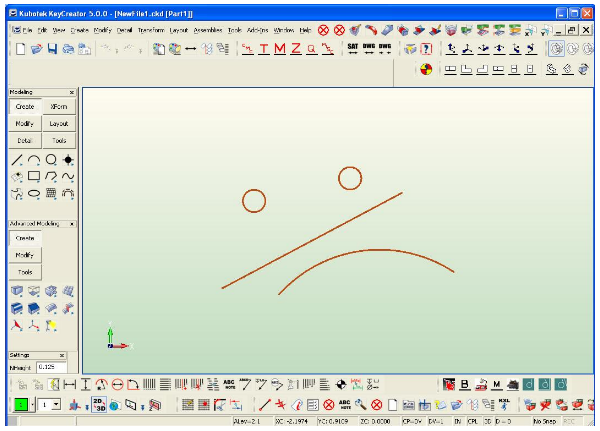
This function is like the auto function in that it will manipulate the display view. The Half function will automatically half the scale of a viewport.

```
Int nVP
SET DISPVIEW, 1, nVP

Auto

Half

EXIT
```

Before



After

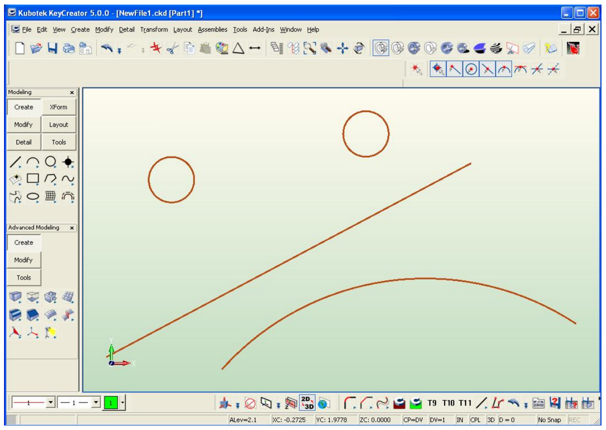


## LevelMod

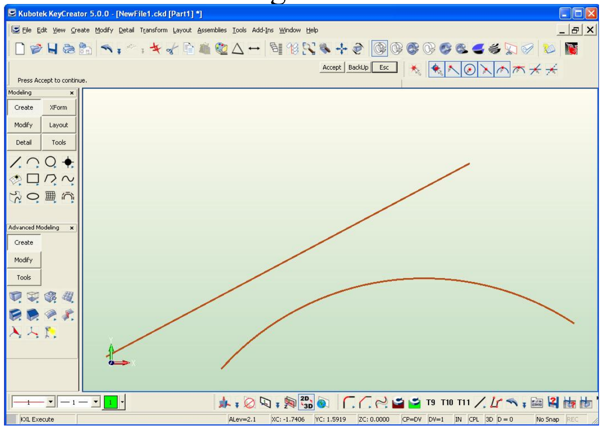
This function sets the attributes of a list of levels or all active mode levels. This example is undisplaying level #1 and then waiting 5 seconds then turning level #1 back on.

```
LEVELMOD nDISP,"1"
notestate
wait 5
LEVELMOD DISP,"1"
notestate
EXIT
```

Before



During the Wait



## LoadDimFile

This function will load a dimension settings file (*.dim) into the active part! This will NOT change dimensions that have already been placed in your drawing.

```
LoadDimFile "C:\\KeyCreator5.0\\Dim File1.dim"

Exit
```

*Note: For this to work correctly you must first create the Dim File1.dim file in the root Keycreator5.0 folder. Again this will NOT change dimensions that are in your drawing it will only work for newly created dimensions.*

To create the dim file you will need to go to the Detail menu bar then go to the settings and again settings. Change the items you need changed for the new file then use the save button when you have changed all of your settings.

## NewLayout

This function will create a new empty layout in the part, with the given parameters.

```
CLEAR hLay12
NEWLAYOUT hLay12, "Test Layout", 17.0000, 11.0000, 1, 2, "",0,1
LOADLAYOUT hLay12

Exit
```

## LoadLayout

This function has two purposes the first is to load a layout in the part window with a predefined layout handle the second is to return from Layout to Model mode.
*For this sample to work correctly you will need to "get" the handle of the layout. Before we get to that function we will show you the basics first. The simple way to do this is to create the layout with a given handle then load the layout with the handle we gave.*

```
//We are using the layout handle form the above example
LOADLAYOUT hLay12

Exit
```

## LoadPart

This is a function that will open the named file/part. If only one part exists in the file then the command will open the file and the part. If there is more than one part in the file and you leave off the second value then you will get a prompt to select the part to open. If there is only one part then the command will open the file and the part.

```
LoadPart "TestFile.ckd", , 1, 0

Exit
```

This will open the file with a part prompt!

---

```
LoadPart "TestFile.ckd", "Part1", 1, 0

Exit
```

This will open the file and the part

## MaskEntity

The MaskEntity command allows for the setting the selection mask to particular entities. This is a replacement for the "SET MASK", and "SET MASKENT" commands. When an entity is masked it is selectable and other entities are not, see below for a list of entity types and their mask flags.

*To remove an entity from a mask set the negative of the entity's number.*

```
MaskCkear
MASKENTITY 2,3

CLEAR hEnt, hEnts, hEntp, hEnti
HENTITY hEnt
INT hEnts
DOUBLE hEntp
HDRAWINST hEnti

GETENT "Select Any Entity", hEnt, hEnti, 0, 0, 1, 0, 0

Exit
```

## MaskClear

This is a function to clear all current selection masks. This is very useful if you have a large KXL program and you have masked in previous commands then you will need to clear the masking.

```
MASKENTITY 2,3

MaskClear

CLEAR hEnt, hEnts, hEntp, hEnti
HENTITY hEnt
INT hEnts
DOUBLE hEntp
HDRAWINST hEnti

GETENT "Select Any Entity", hEnt, hEnti, 0, 0, 1, 0, 0

Exit
```

*In this sample you see that you can select any entity even though you masked for entities number 2,3 if the maskClear was above the MaskEntity 2,3 then you would only be able to select entity types 2 and 3.*

## NewPart

This function has two functions, it can create a new design file or part within an existing design file.

```
NewPart "TestFile.ckd", 0, "New Part Test", 1

Exit
```

*This sample will place a new part in the active file of TestFile.ckd, the part name will be the New Part Test. It will also load and display the part if you do not want the new part displayed the last variable needs to be replaced with a "0" and it will be created but not shown.*

## NoteState

This is a function that you will be very familiar with! This will place a notestate in the database. This should be used after an action that causes the database to be altered. Geometric entities that are created may not display until a NoteState command has been issued.

```
LEVELMOD nDISP,"1"
notestate
Wait 5
LEVELMOD DISP,"1"
notestate
EXIT
```

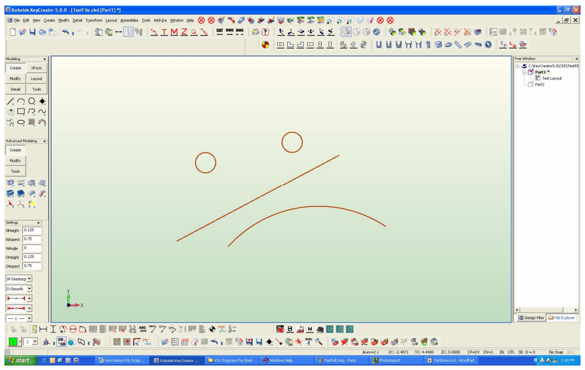
This will also set an undo/redo state as well so if you create several items at once then notestate after all are creates then the undo will undo all items.

## PanDown

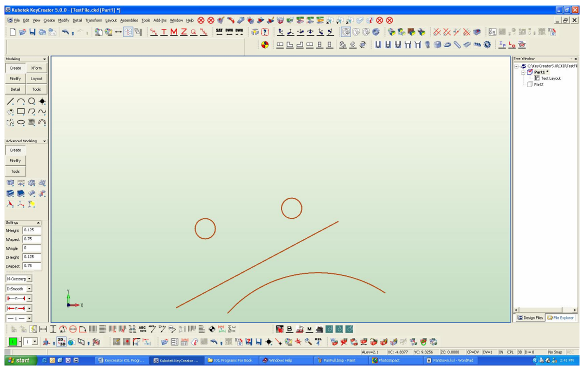
This is a simple command that will pan a viewport DOWN by a percent of the view port. The amount is based on the view port size as the sample shows the .25 will be 25% of the view port size to be paned.

```
PanDown .25, 0

Exit
```

Before



After

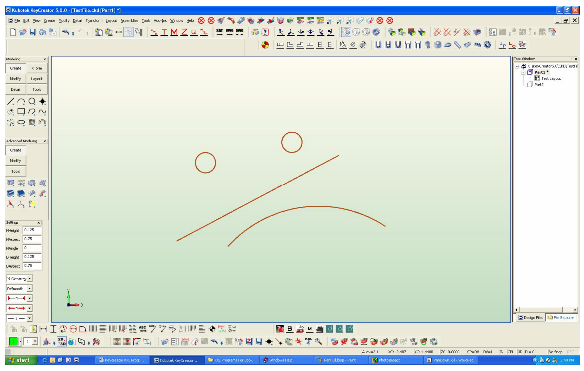


## PanLeft

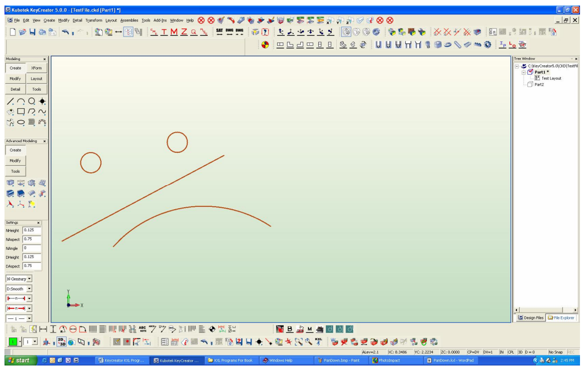
This is a simple command that will pan a viewport LEFT by a percent of the view port. The amount is based on the view port size as the sample shows the .25 will be 25% of the view port size to be paned.

```
Panleft .25, 0

Exit
```

Before



After

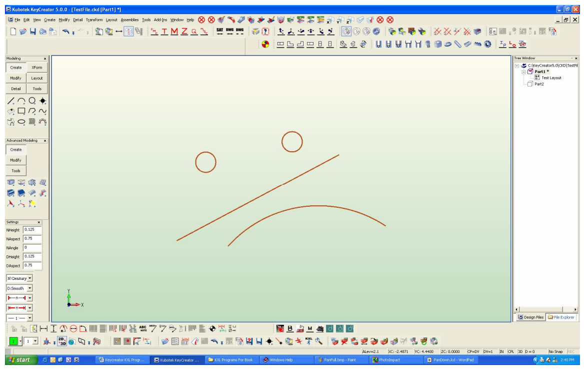


## PanRight

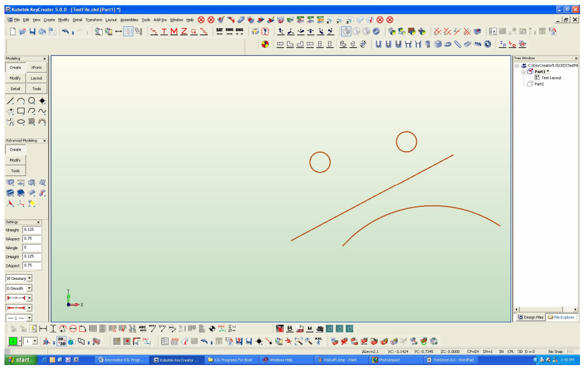
This is a simple command that will pan a viewport RIGHT by a percent of the view port. The amount is based on the view port size as the sample shows the .25 will be 25% of the view port size to be paned.

```
PanRight .25, 0

Exit
```

Before



After

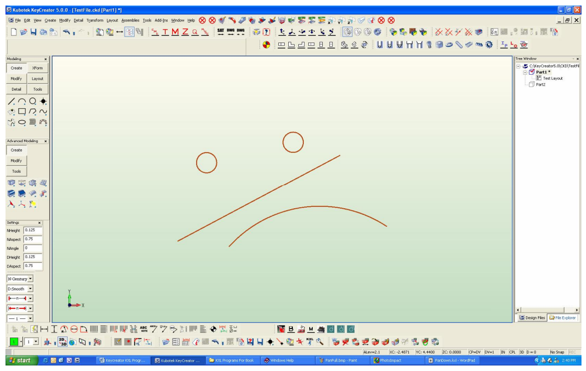


## PanUp

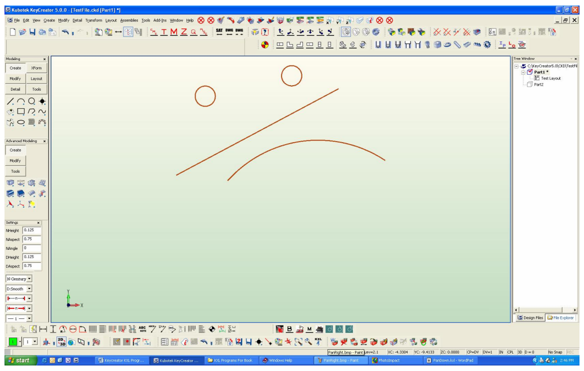
This is a simple command that will pan a viewport UP by a percent of the view port. The amount is based on the view port size as the sample shows the .25 will be 25% of the view port size to be paned.

```
Panup .25, 0

Exit
```

Before



After



## PrintDirect

Sometimes you will need to print using the KXL functions. This function will start the print interaction using the given viewport. You can use the active display view port or a predefined display view port.

```
PrintDirect 0

Exit
```

This function will use the default printer and the default KeyCreator printing setup in the Tool> Options> Plotting/Printing Tab. This will Print the "0" view port (Active View Port).

## PrintPreview

This will let you preview your print with the option to select the printer and printer options and let you print after the preview. This function will start the print interaction using the given viewport. You can use the active display view port or a predefined display view port.

```
PrintPreview 0

Exit
```

## PrintVPort

This function is a combination of the previous two commands in that it will let you print with the option to select the printer and printer options. This function will start the print interaction using the given viewport. You can use the active display view port or a predefined display view port.

```
PrintPreview 0

Exit
```

## Prompt

Displays a text string on the prompt line but does not pause for user input. A format control string must be supplied, additional value parameters are optional. The resulting text string must not exceed 68 characters or truncation will occur.

```
Pause"This will test the prompt command"

Double dTime
dTime = 10

Prompt "Hello There! This will be displayed for %d
seconds", dTime

Wait dTime
Exit
```

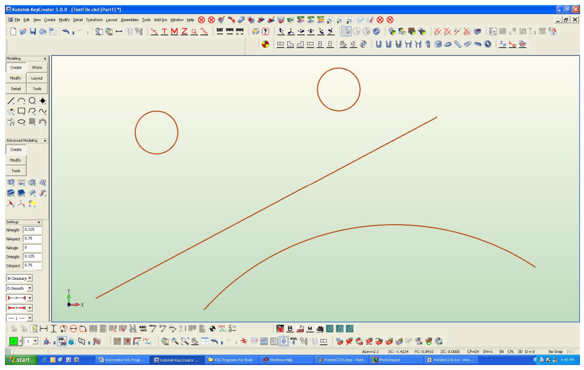
You can use this as a prompt if your functions are going to take awhile to complete and still let the user know what is going on.

## RotateCCW

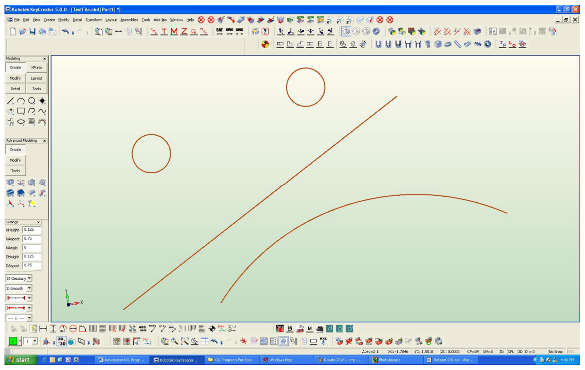
This function will rotate a viewport COUNTER-CLOCK-WISE! This is a simple function just like the pan functions. This is very useful in KXL's that will "Fly" through or around your part with one button click.

```
RotateCCW 10, 0

Exit
```

Before



After



*Note:*

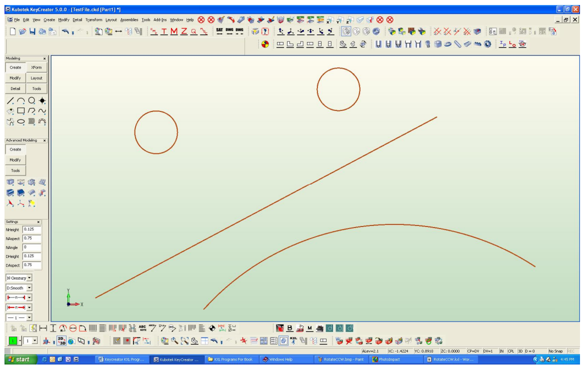
*One thing to remember this will rotate in the Zaxis of the view port. The first variable is the angle to rotate the view by and the last one is the view port number!*

## RotateCW

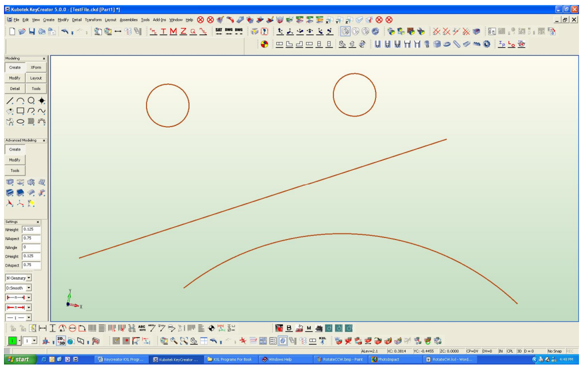
This function will rotate a viewport CLOCK-WISE! This is a simple function just like the pan functions. This is very useful in KXL's that will "Fly" through or around your part with one button click.

```
RotateCW 10, 0

Exit
```

Before



After



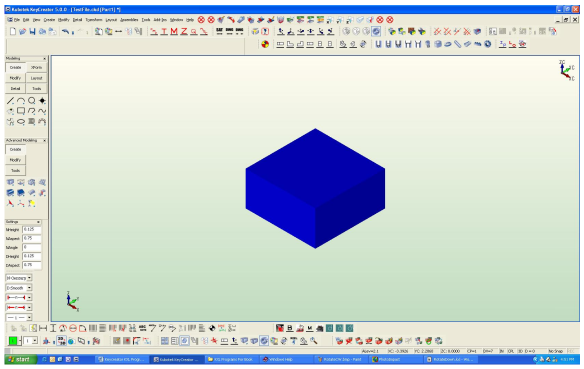
*Note:*
*One thing to remember this will rotate in the Zaxis of the view port. The first variable is the angle to rotate the view by and the last one is the view port number!*

## RotateDown

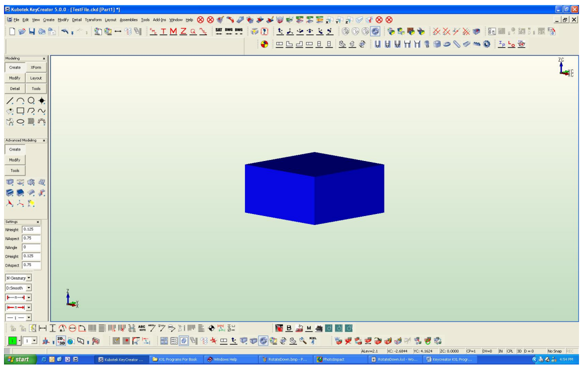
This function will rotate a viewport DOWN! This is a simple function just like the pan functions. This is very useful in KXL's that will "Fly" through or around your part with one button click.

```
RotateDown 25, 0

Exit
```

Before



After



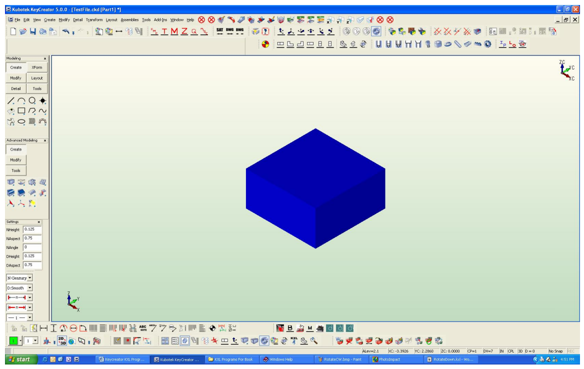
*Note:*
*One thing to remember this will rotate in the Xaxis of the view port. The first variable is the angle to rotate the view by and the last one is the view port number!*

## RotateLeft

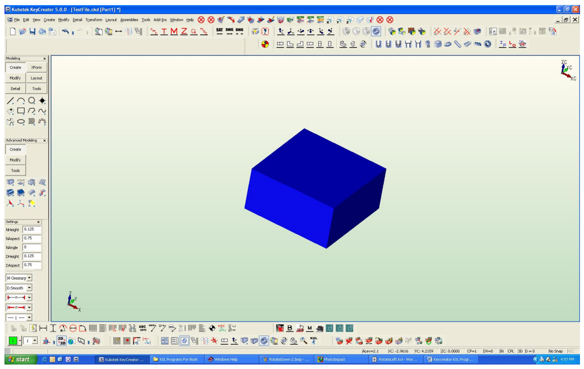
This function will rotate a viewport to the LEFT! This is a simple function just like the pan functions. This is very useful in KXL's that will "Fly" through or around your part with one button click.

```
RotateLeft 15, 0

Exit
```

Before



After



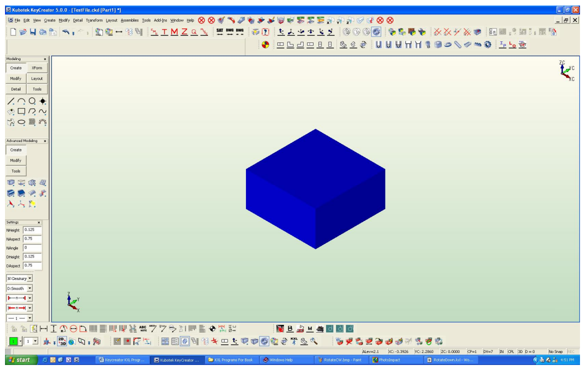
*Note:*
*One thing to remember this will rotate in the Yaxis of the view port. The first variable is the angle to rotate the view by and the last one is the view port number!*

## RotateRight

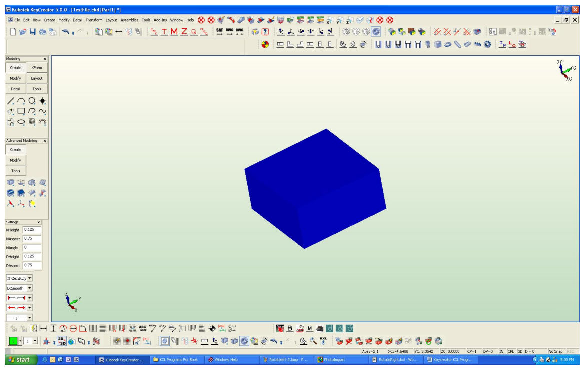
This function will rotate a viewport to the RIGHT! This is a simple function just like the pan functions. This is very useful in KXL's that will "Fly" through or around your part with one button click.

```
RotateRight 15, 0

Exit
```

Before



After



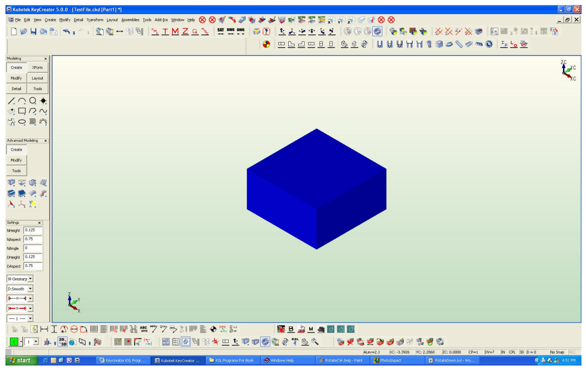
*Note:*
*One thing to remember this will rotate in the Yaxis of the view port. The first variable is the angle to rotate the view by and the last one is the view port number!*

## RotateUp

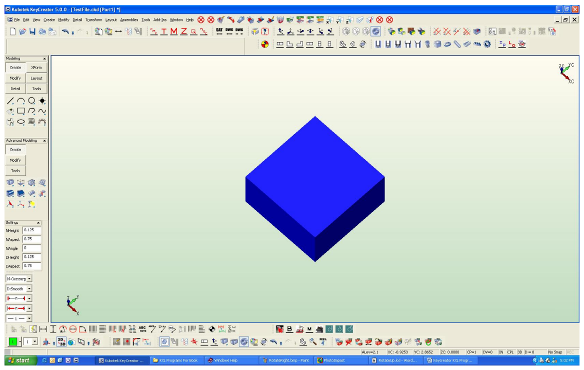
This function will rotate a viewport UP! This is a simple function just like the pan functions. This is very useful in KXL's that will "Fly" through or around your part with one button click.

```
RotateUp 25, 0

Exit
```

Before



After



*Note:*
*One thing to remember this will rotate in the Xaxis of the view port. The first variable is the angle to rotate the view by and the last one is the view port number!*

## SavePart

This function serves as two functions in one, first it can save the active file, second it can optionally save with a different filename.

```
SavePart

Exit
```

This will save the active part!

```
SavePart "New Part2.ckd"

Exit
```

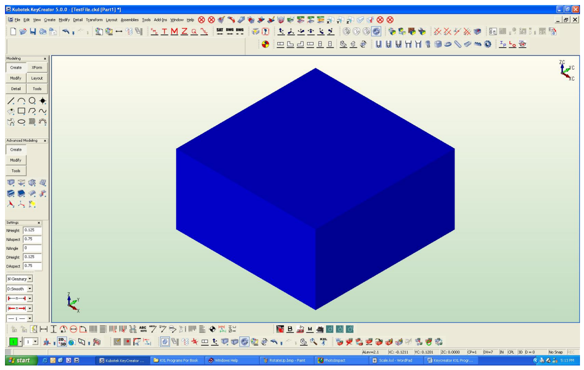
This will Save the active part with a new name!

## Scale

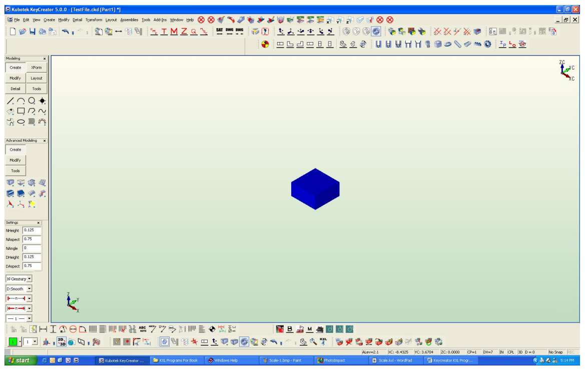
Rescales the system part displayed. This performs the same function as using the system's Immediate Mode command, or activating the Scale option in the Status Window.

```
Scale 0, .5

Exit
```

Before



After



This function can also scale the view port from a location. Simply place the locations to scale from, in the last two variable fields.

## UpdateDetailEnts

This is a simple function that has no variables or other formats. This will update the attributes of the detail entities in the selection list to the currently active settings of the part. Just this one line!

*UpdateDetailEnts*

## Wait

As you have seen in some examples in this book this function will suspend a KXL file execution for a specified number of seconds. This is very useful if you need to wait for an external function in another program such as executing the "DateTime.exe" program that will write the date and time to a file then WAIT for the file then open it and read the file into KXL variables.

```
Pause"This will test the prompt command"

Double dTime
dTime = 10

Prompt "Hello There! This will be displayed for %d seconds", dTime

Wait dTime
Exit
```

In this sample, the same as the prompt command, I am showing that you can use a value or a variable for the wait command.