



# 18

## Customizing MicroStation Descartes

This chapter explains how to customize MicroStation Descartes: how to use the workspace editing capabilities of MicroStation to create workspaces by yourself; how to build your own tool boxes or tool frames by picking any tool from MicroStation Descartes or from MicroStation itself; how to use the MicroStation BASIC macro recorder and the MicroStation Descartes key-ins to build your own commands or applications.

---

### About Customization

MicroStation Descartes offers an extensive set of more than 1000 commands that allows full customization and automation using either the MicroStation BASIC or the MDL programming language. In MicroStation Descartes, all commands are accessible in a consistent manner using key-ins. Even though key-ins can be typed in a command line or used in customized workspaces, in scripts or in UCM, the real power will be achieved when called from a MicroStation BASIC or MDL application. An application based on such key-ins will not need to be recompiled or adapted when new versions of Descartes become available. The complete list of Descartes key-ins is presented in Appendix A.

Descartes key-ins follow a standard nomenclature to help you find them or even deduce them easily and quickly. Descartes key-ins are made of three words followed by optional parameters,

each word having its own meaning as shown in the following table.

<b>Module</b>	<b>Object</b>	<b>Action</b>
<p>Related key-ins are grouped in modules. This modular approach serves as a quick index for finding related key-ins. The module name always starts with IMG, this groups all Descartes functions together in the Key-in Browser.</p> <p>Examples are:            IMG            IMGMANAGER            IMGEDIT            IMGTRANSFORM            IMGREGISTER            ...</p>	<p>The object on which the action will be performed. These objects can be <i>dialog boxes, tool boxes, tools, files, views, images, color filters</i>, etc.</p> <p>Examples are:            DIALOG            DLG<i>name</i>            TOOL<i>name</i>            PROJECT            FILE            FILTER            STAMP            ...</p>	<p>Specifies the action done with the object. If the object is a dialog box, tool box, file or project, it can be <i>opened</i> or <i>closed</i>. If the object is a variable, typically presented in a dialog box item, a <i>set</i> and a <i>get</i> action is available.</p> <p>Examples are:            OPEN            CLOSE            DELETE            SET<i>something</i>            GET<i>something</i>            ...</p>

Development of a solution can be as simple as a macro and as efficient as an application intended for a broad audience of users. In a typical development process today, it is common practice to structure the application into smaller tasks that are quickly prototyped with visual tools (example: the MicroStation BASIC recorder) to verify the satisfaction of user requirements. You can then try, comment and iterate until the right functionality is obtained. The prototype can then evolve until you build a complete application, or be thrown away for a complete re-implementation in the appropriate language.

## Customizing the Workspace

All Descartes commands can be integrated in user-defined menus, tool boxes and tool frames with the MicroStation Workspace > Customize tool. If Tool Boxes or Tool Frames is selected in the upper left option button in the Customize dialog box, the Descartes tool boxes are included in the list of Available Tools, and they can be copied in existing or new tool boxes and tool frames. If Menu Bar is selected, new menus and menu items can be added to provide access to the appropriate Descartes command with a key-in. Once a customized workspace is saved, you can use it repeatedly, and you can even pass it to other users. For more information on workspaces, refer to the *MicroStation Administrator's Guide*.

## CAD Overlay Workspace

An additional workspace, "Cadovl," is delivered with MicroStation Descartes Version 7. This workspace is intended to offer an easy migration for AutoCAD/CAD Overlay users to MicroStation/MicroStation Descartes Version 7.

### CAD Overlay Image Menu

This workspace is made from the default Microstation workspace. It adds to it the Image Menu from which MicroStation Descartes functions can be accessed. Figure 21 displays the Microstation SE menu including the Image menu. Figure 22 displays the Image menu.



Figure 22



Figure 21

A majority of the CAD Overlay menu items have corresponding MicroStation Descartes function. The following is a cross-reference between CAD Overlay menu items and MicroStation Descartes.

#### Insert

Opens an existing image file.

Descartes: File/Open

**Export > Save As...**

Saves an opened image using a new name.

Descartes: File/Save As...

**Export > BatchConvert**

Opens the dialog box that allows to convert image files.

Descartes: File/Batch Conversion...

**Close**

Close an opened image.

Descartes: File/Close

**Close All**

Closes all opened images.

Descartes: File/Close All

**Manage...**

Opens the Image Manager main dialog box.

MicroStation: File/Dcartes - Image Manager

**Image > Select/Active**

Starts the Image Manager tool that allows image selection.

Descartes: Prepare tool bar: Select/Active tool

**Image > Display On**

Turns on image displaying in a view.

Descartes: Display/Images On

**Image > Display Off**

Turns off image displaying in a view.

Descartes: Display/Images Off

**Image > Bring To Front**

Brings a specific image on top of a view.

Descartes: Display/Bring To Front

**Image > Send To Back**

Send a specific image on back of a view.

Descartes: Display/Send To Back

### **Image > Fit Image To View**

Displays the specified images at a scale such that they fit in the view.

Descartes: Display/Fit Image to View

### **Image > Fit All Images To View**

Displays all visible images from the list associated with the view within the display window for the specified view.

Descartes: Display/Fit All Images to View

### **Image > Normal (1:1)**

Adjusts the scale of the view and displays the specified image without loss of resolution or zoom. This image displays with its actual resolution of 1:1 and the other images adjust accordingly.

Descartes: Display/Actual Resolution (1:1)

### **Correlate > Manual**

Starts the Image Transform Modify Image tool allowing image moving, scaling, rotation and affinity changes.

Descartes: Image Transform tool bar: Modify Image tool

### **Correlate > Move**

Starts the Image Transform Move Image tool.

Descartes: Image Transform tool bar: Move tool

### **Correlate > Scale**

Starts the Image Transform Scale Image tool.

Descartes: Image Transform tool bar: Scale tool

### **Correlate > Match**

Starts the Image Transform Warp Image tool and sets it to do a similitude transformation.

Descartes: Image Transform tool bar: Warp tool

### **Correlate > Deskew**

Starts the Image Transform Square Image tool and sets it to square a diagonal axis.

Descartes: Image Transform tool bar: Square tool

**Correlate > Helmert**

Starts the Image Transform Warp Image tool and sets it to do a Helmert transformation.

Descartes: Image Transform tool bar: Warp tool

**Correlate > Affine**

Starts the Image Transform Warp Image tool and sets it to do an affine transformation.

Descartes: Image Transform tool bar: Warp tool

**Correlate > Rubbersheet**

Starts the Image Transform Warp Image tool and sets it to do a projective transformation.

Descartes: Image Transform tool bar: Warp tool

**Correlate > Register**

Opens the Register dialog box.

Descartes: Utilities/Register...

**Properties > Color**

Opens the Color Table Editor dialog box.

Descartes: File/Properties/Color Table...

**Properties > Transparent Colors**

Opens the Transparent Colors dialog box.

Descartes: Utilities/Transparent Colors...

**Properties > Translucent Colors**

Opens the Translucent Colors dialog box.

Descartes: Utilities/Translucent Colors...

**Properties > Correlation Info**

Opens the Image Location Editor dialog box.

Descartes: File/Properties/Location...

**Properties > File Info**

Opens the General Image Properties dialog box.

Descartes: File/Properties/General...

**Configure > Settings > Clipboard**

Opens the Copy Image Settings dialog box.

Descartes: Copy/Paste tool bar: Copy Image Settings tool

**Configure > Settings > Theme Styler**

Opens the Theme Styler dialog box.

Descartes: Theme tool bar: Theme Styler Dialog tool

**Configure > Settings > Undo**

Opens the Undo Image Settings dialog box.

Descartes: Undo/Redo tool bar: Undo Image Settings tool

**Configure > Settings > Vector Merging**

Opens the Stamp Vector Settings dialog box.

Descartes: Stamp Vector tool bar: Stamp Vector Settings tool

**Configure > Settings > Vtool**

Opens the Vectorize Settings dialog box.

Descartes: Vectorize tool bar: Vectorize Settings tool

**Configure > Memory**

Opens the Memory Allocation dialog box.

Descartes: Settings/Preferences/Memory...

**Configure > Color Distribution**

Opens the Color Distribution dialog box.

Descartes: Settings/Preferences/Color Distribution...

**Filters > Histogram**

Opens the Histogram Display dialog box.

Descartes: File/Properties/ Histogram...

**Filters > Color Compression**

Opens the Color Compression dialog box.

Descartes: Utilities/Color Compression...

**Filters > Contrast/Brightness**

Opens the Contrast/Brightness dialog box.

Descartes: Display/Contrast/Brightness...

**Filters > Contrast Stretch**

Opens the Contrast Stretch dialog box.

Descartes: Utilities/Contrast Stretch...

**Filters > Density Slicing**

Opens the Density Slicing dialog box.

Descartes: Utilities/Density Slicing...

**Filters > Invert**

Inverts every color of the current color table.

Descartes: File/Properties/Color Table...

**Filters > Despeckle > Measure**

Starts the Clean Up Measure Speckle tool.

Descartes: Clean Up tool bar: Measure Speckle tool

**Filters > Despeckle > Remove**

Starts the Clean Up Remove Speckle - Area tool.

Descartes: Clean Up tool bar: Remove Speckle - Area tool

**Filters > Despeckle > Fill**

Starts the Clean Up Remove Holes - Area tool.

Descartes: Clean Up tool bar: Remove Holes - Area tool

**Filters > Mirror**

Starts the Image Transform Mirror Image tool.

Descartes: Image Transform tool bar: Mirror Image tool

**Display Frames > On**

Turn on all image frames.

Descartes: Settings/View Image Attributes/Image Border

**Display Frames > Off**

Turn off all image frames.

Descartes: Settings/View Image Attributes/Image Border

**CAD as Raster Editor**

Opens the Image Lock dialog box.

Descartes: Settings/Lock...



### **Snap On Tentative**

Opens the Raster Snap on Tentative dialog box.

Descartes: Settings/Raster Snap on Tentative...

### **Color Filters > Add Color**

Starts the Color Filter Add to Filter tool.

Descartes: Color Filter tool bar: Add to Filter tool

### **Color Filters > Subtract Color**

Starts the Color Filter Subtract from Filter tool.

Descartes: Color Filter tool bar: Subtract from Filter tool

### **Color Filters > Hilite**

Starts the Color Filter Highlight Filter tool.

Descartes: Color Filter tool bar: Highlight Filter tool

### **Color Filters > Dialog**

Opens the Color Filter dialog box.

Descartes: Color Filter tool bar: Color Filter Dialog tool

### **Undo**

Undo last action.

Descartes: Undo/Redo tool bar: Undo Image tool

### **Redo**

Redo last action.

Descartes: Undo/Redo tool bar: Redo Image tool

### **Vtool > Line/Pline/Arc**

Starts the Image Vectorize Place Element tool.

Descartes: Image Vectorize tool bar: Place Element tool

### **Vtool > Continue**

Starts the Image Vectorize Continue Element tool.

Descartes: Image Vectorize tool bar: Continue Element tool

### **Vtool > Insert**

Starts the Image Vectorize Insert Element tool.

Descartes: Image Vectorize tool bar: Insert Element tool

**Vtool > Text > Select Horizontal**

Starts the Text/Cells Select Horizontal Text tool.

Descartes: Text/Cells tool bar: Select Horizontal Text tool

**Vtool > Text > Select Oriented**

Starts the Text/Cells Select Oriented Text tool.

Descartes: Text/Cells tool bar: Select Oriented Text tool

**Vtool > Text > Convert Text**

Starts the Text/Cells Convert Text tool.

Descartes: Text/Cells tool bar: Convert Text tool

**Vtool > Cells > Convert**

Starts the Text/Cells Convert Cell tool.

Descartes: Text/Cells tool bar: Convert Cell tool

**Vtool > Cells > Convert Horizontal**

Starts the Text/Cells Convert Horizontal Cell tool.

Descartes: Text/Cells tool bar: Convert Horizontal Cell tool

**Vtool > Cells > Convert Oriented**

Starts the Text/Cells Convert Oriented Cell tool.

Descartes: Text/Cells tool bar: Convert Oriented Cell tool

**Vtool > Follow > Work Area > Define**

Starts the Image Vectorize Place Nodes tool.

Descartes: Image Vectorize tool bar: Place Nodes tool

**Vtool > Follow > Work Area > Remove**

Starts the Image Vectorize Delete Nodes tool.

Descartes: Image Vectorize tool bar: Delete Nodes tool

**Vtool > Follow > Lines**

Starts the Image Vectorize Convert Lines tool.

Descartes: Image Vectorize tool bar: Convert Lines tool

**Vtool > Follow > All Connected Lines**

Starts the Image Vectorize Convert All Connected Lines tool.

Descartes: Image Vectorize tool bar: Convert All Connected Lines tool

**Vtool > Follow > Contour Lines**

Starts the Image Vectorize Convert Contour Lines tool.

Descartes: Image Vectorize tool bar: Convert Contour Lines tool

**Vtool > Follow > Circle**

Starts the Image Vectorize Convert Circle tool.

Descartes: Image Vectorize tool bar: Convert Circle tool

**REM > Copy to Clipboard**

Starts the Copy/Paste Copy Image Area tool.

Descartes: Copy/Paste tool bar: Copy Image Area tool

**REM > Paste from Clipboard**

Starts the Copy/Paste Paste Image Area tool.

Descartes: Copy/Paste tool bar: Paste Image Area tool

**REM > Move Area**

Starts the Local Transform Move Image Area tool.

Descartes: Local Transform tool bar: Move Image Area tool

**REM > Scale Area**

Starts the Local Transform Scale Image Area tool.

Descartes: Local Transform tool bar: Scale Image Area tool

**REM > Rotate Area**

Starts the Local Transform Rotate Image Area tool.

Descartes: Local Transform tool bar: Rotate Image Area tool

**REM > Mirror Horizontal Area**

Starts the Local Transform Horizontal Mirror Image Area tool.

Descartes: Local Transform tool bar: Horizontal Mirror Image Area tool

**REM > Mirror Vertical Area**

Starts the Local Transform Vertical Mirror Image Area tool.

Descartes: Local Transform tool bar: Vertical Mirror Image Area tool

**REM > Transform Area**

Starts the Local Transform Place Image Area tool.

Descartes: Local Transform tool bar: Place Image Area tool

**Paint > Rectangular Brush**

Starts the Paint Rectangular Brush tool.

Descartes: Paint tool bar: Paint with Rectangular Brush tool

**Paint > Circle Brush**

Starts the Paint Circular Brush tool.

Descartes: Paint tool bar: Paint with Circular Brush tool

**Paint > Oblique Brush**

Starts the Paint Oblique Brush tool.

Descartes: Paint tool bar: Paint with Oblique Brush tool

**Paint > Area**

Starts the Paint Area tool.

Descartes: Paint tool bar: Paint Area tool

**Rub > Polyline**

Starts a line shaped rubber (eraser) function. Once started, this command allows user to trace a polyline using standard vectorize function. Once the line is drawn, the command erases the raster element being under the line.

Descartes: Place Element with Erase Raster ON in *point* mode.

**Rub > Arc**

Starts an arc shaped rubber (eraser) function. Once started, this command allows user to trace an arc using standard vectorize function. Once the line is drawn, the command erases the raster element being under the line.

Descartes: Place Element with Erase Raster ON in *arc* mode.

**Rub > Rectangular Brush**

Starts the Erase Rectangular Brush tool.

Descartes: Erase tool bar: Erase with Rectangular Brush tool

**Rub > Circle Brush**

Starts the Erase Circular Brush tool.

Descartes: Erase tool bar: Erase with Circular Brush tool

## **Rub > Oblique Brush**

Starts the Erase Oblique Brush tool.

Descartes: Erase tool bar: Erase with Oblique Brush tool

## **Rub > Area**

Starts the Erase Area tool.

Descartes: Erase tool bar: Erase Area tool

## **Merge > Merge Vectors**

Starts the Stamp Vector tool.

Descartes: Stamp Vector tool bar: Stamp Vector tool

## **Merge > Merge Images**

Starts the Image Transform Merge Images tool.

Descartes: Image Transform tool bar: Merge Images tool

## **Plot**

Opens the MicroStation Plot dialog box.

MicroStation: File/Print/Plot

## **Help**

Opens Descartes User's Guide window.

Descartes: Help/Contents

## **About....**

Opens Descartes About dialog box.

Descartes: Help/About Descartes...

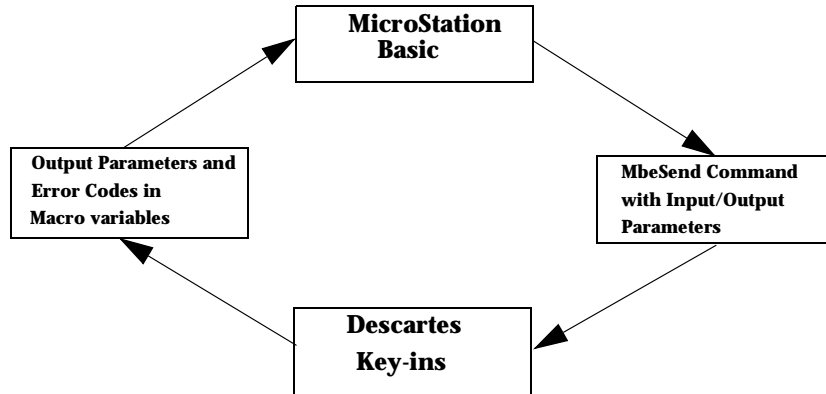
## **Programming with BASIC macros**

The MicroStation BASIC macro recorder is the perfect tool to build simple commands or even prototypes that will evolve to a professional application. In MicroStation BASIC, the function MbeSend Command should be used to call the desired key-in.

For example, to open the Image File tool box, use:

MbeSend Command "IMGMANAGER TOOLFILE OPEN"

Many Descartes key-ins accept input and output parameters, and all of them return error codes to allow full control of the flow of the program when errors occur. The following diagram shows how parameters and error codes are handled.



All input parameters are passed by value directly. Returned values and error codes will be passed through MicroStation BASIC public variables. The MbeSend Command function accepts a string parameter containing the key-in command and the optional input or output parameters.

For example, to open an image, the image file path is given as an input parameter:

```
MbeSend Command "IMGMANAGER FILE OPEN c:\example.hmr"
```

The following key-in returns the X coordinate of the origin of the active image in an output parameter who is defined as the BASIC public variable "originx."

```
MbeSend Command "IMGMANAGER LOCATION GETXORIGIN originx"
```

In the next example, the key-in has an input parameter (the view number 1) and an output parameter (the BASIC public variable "status"), the latter receives the status of images (on/off) in the specified view.

```
MbeSend Command "IMGMANAGER VIEWVISIBLE GET 1 status"
```

All Descartes commands return an error code that can be verified in the flow of the program. Note that this only applies to Descartes key-ins, that begin with the letters IMG. The programmer must declare the BASIC public variable with "Public ErrorCode as Long" as demonstrated in the next source code example. When the command is executed with the MbeSendCommand, Descartes commands assign to the variable ErrorCode the appropriate error code, which will be KIN SUCCESS

for success and other values for different possible errors. An error code verification is shown in the next example:

```
' Open the image c:\example.hmr
Public ErrorCode as Long
Sub main
MbeSendCommand "IMGMANAGER FILE OPEN c:\example.hmr"
  If ErrorCode <> KIN SUCCESS Then
    MbeMessageBox "Error MicroStation Descartes..."
  End If
End Sub
```

- 📌 KIN SUCCESS should be defined as a constant of value 0. A sample of all possible error codes is provided in the example source file "... \dcartes\macros\imgmain.bas."

## Programming with MDL

If you want to have an application that is rich in functionality and uses some features not supported in MicroStation BASIC, like the utilization of modeless dialog boxes and the creation of new elements, you will have access to a much broader extent of possibilities with the MDL language. Descartes commands can also be called in MDL with the "mdlInput sendKeyin()" function. However, Descartes provides an equivalent improved function "imgInput callKeyin ()" that has the advantages of handling Descartes error codes and that executes commands synchronously. Effectively, "mdlInput sendKeyin()" sends a key-in to the input queue of MicroStation and the key-in only processes when the flow permits. Meanwhile, the MDL program continues to the next instruction without knowing if the sent key-in has been processed. To work around this problem of asynchronism, the Descartes "imgInput callKeyin ()" function executes the given key-in immediately and handles the error codes in the specified "tcb-> uc xx" variable.

"imgInput callKeyin ()" is not a function available in a static or dynamic library. The pointer to this function can be obtained at run-time using another function named "imgInput startKeyinCall ()". The "imgInput startKeyinCall()" function behaves in a similar way to the "mdlState startXXX()" function. It takes as a parameter, a pointer to a callback function which will be called when the work is done; in this case, when the public variable "imgInput callKeyin" is assigned. A complete example of source code is provided in the "... \dcartes\mdl\examples\imgmain" directory. The programmer should pay attention to the "imgmain.mc" source file. This example

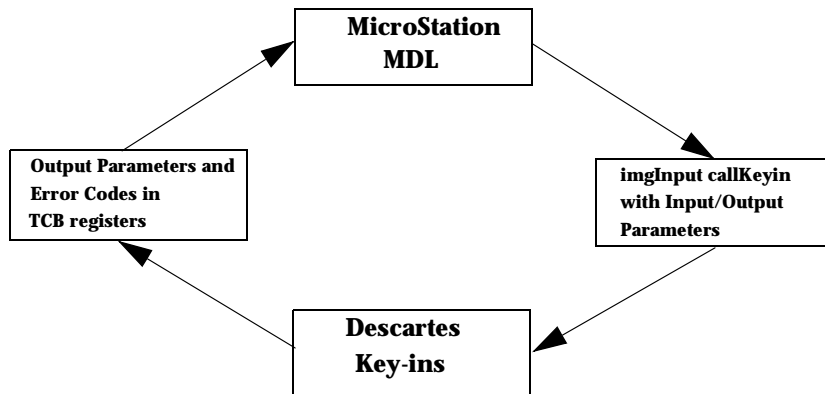
can be customized to specific needs. All other files in this directory beginning with “img” should be compiled with the user application; a make file is provided.

Once “imgInput callKeyin” is assigned (this needs to be done only once in an application), it can be used to call Descartes key-ins. For example, to open the Image File tool box:

```
imgInput callKeyin (“IMGMANAGER TOOLFILE OPEN”);
```

Many Descartes key-ins accept input and output parameters, and all of them return error codes to allow full control of the flow of the program when errors occur.

The following diagram shows how parameters and error codes are handled.



All input parameters are passed by value directly. Returned values and error codes will be passed through TCB registers. The “imgInput callKeyin” function accepts a string parameter containing the key-in command and the optional input or output parameters.

For example, to open an image, the image file path is given as an input parameter:

```
imgInput callKeyin (“IMGMANAGER FILE OPEN c:\example.hmr”);
```

The following key-in returns the X coordinate of the origin of the active image in an output parameter which is defined as a tcb register number.

```
imgInput callKeyin (“IMGMANAGER LOCATION GETXORIGIN 2”);
```

The output parameter “2” is the tcb register number that receives the X coordinate of the origin of the active image. In the following example the output type is a string, which means the tcb register type receiving the output value will be the character variable “tcb->uc c[2]”. As a help, some definitions are provided in “imginput.h” to access tcb registers. For example, REG STRING CONT(2) can be used to access “tcb->uc c[2]”.



In this example, the key-in has an input parameter (the view number 1) and an output parameter (tcb->uc c[2]): the latter receives the status of images (on/off) in the specified view.

```
imgInput callKeyin ("IMGMANAGER VIEWVISIBLE GET 1 2");
```

All Descartes commands return an error code that can be verified in the flow of the program. Note that this only applies to Descartes key-ins, begin with the letters IMG. The error code is always returned in the "tcb->uc i[15]" register (the definition "#define ERROR CODE (tcb->uc i[15])" is made in `imginput.h`). When the command is executed with the `imgInput callKeyin` function, Descartes commands assign to ERROR CODE the appropriate error code, which will be KIN SUCCESS for success and other values for different possible errors (see `imginput.h`). An error code verification is shown in the next example:

```
#include "imginput.h"
myfunction()
{
/* Open the image c:\example.hmr */
imgInput callKeyin ("IMGMANAGER FILE OPEN c:\example.hmr");
if (ERROR CODE!= KIN SUCCESS)
printf("Error in...");
}
```

For a complete example with different types of output parameters, refer to the example in the "...\\dcartes\\mdl\\examples\\imgmain" directory. All MicroStation Descartes key-ins are listed in Appendix A.

