PictContainer Plug-In

By Thomas Maul, General Manager, 4D Germany.

Technical Note 06-24

Introduction

This Technical Note introduces a 4D plug-in that provides enhanced support for picture handling with 4th Dimension (4D).

With this plug-in pictures can be read, written, and displayed (with scroll bars and support for zooming).

The plug-in uses the GDI+ API on Windows and the QuickTime API on Mac.

The C source code of the plug-in is included with this Technical Note, along with a sample database that demonstrates the plug-in features.

Main Features

OS native library

The PictContainer plug-in uses QuickTime on Mac OS, which is the preferred library for picture management from Apple and comes preinstalled on all Macintosh computers. On Windows it uses GDI+, the current graphical API from Microsoft. GDI+ is preinstalled on Windows XP, Windows 2003 and future systems. Note: for information on Windows 2000 support see the section titled “GDI+ for Windows 2000” at the end of this document.

GDI+ is used on Windows instead of QuickTime because the chances are better that GDI+ is already installed. Additionally the GDI+ Dynamic Link Library (DLL) can be shipped with the 4D application, eliminating the need for special installation requirements. The handling of damaged pictures is better with GDI+ on Windows as well.

Scroll bar support

Depending of the size of the picture, the available space in the plug-in area, and the zoom settings the plug-in automatically displays or hides scroll bars.

Drag & Drop support

The PictContainer plug-in area supports Drag & Drop of pictures from 4D picture variables, fields, or scrollable areas automatically. Dragging from a PictContainer area to other 4D objects can be supported by programming.

Supported picture formats

Mac OS

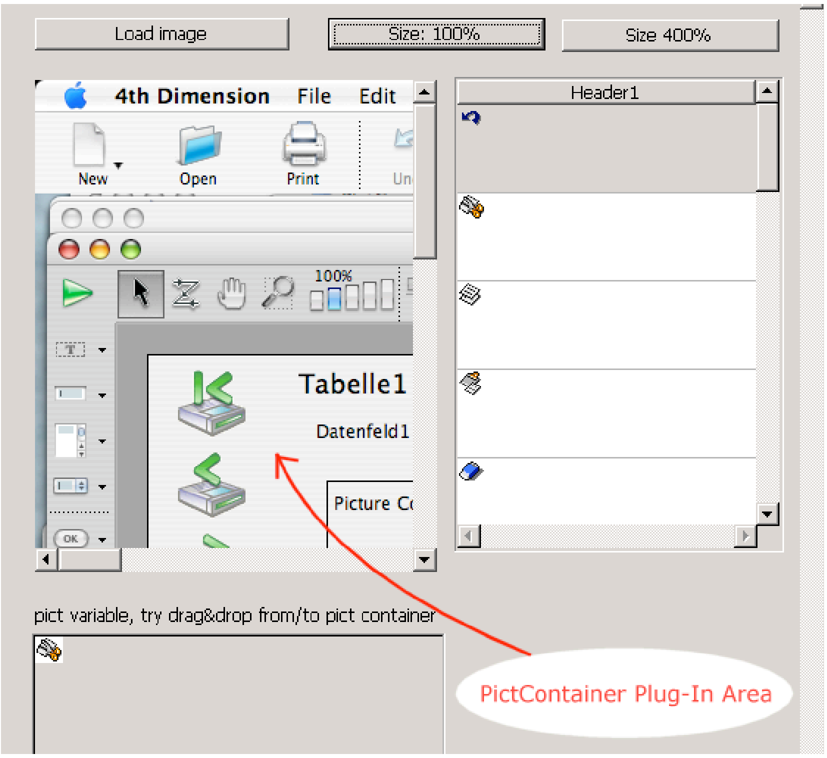
- For reading: All image formats from QuickTime are supported. These are the same formats that are available from the picture commands in 4D, e.g. Pict, BMP, GIF, JPG, PNG, TIFF, Photoshop, etc.
- For writing: The plug-in converts all opened documents to Macintosh pictures (similar to what the 4D picture commands do), so it is suggested to store them in picture variables or keep the original document. The plug-in is not written to support converting pictures, so the existing 4D commands should be used (e.g. WRITE PICTURE FILE or PICTURE TO BLOB). The plug-in contains code to write pictures as BMP, JPG, PNG and TIFF (TIFF always without compression).

Windows

- For reading: BMP, GIF, JPG, PNG, TIFF, EMF, WMF, ICO
- For writing: BMP, GIF, JPG, PNG, TIFF

Using the PictContainer Plug-In

The example database included with this Technical Note demonstrates some of the features available in the PictContainer plug-in. The database is called "PictContainer.4DB". It starts in User Mode and contains 1 table with 1 record. The input for the [Table1] table is shown below:



With this input form the following tasks can performed:

- Load an image from disk into the PictContainer plug-in area.
- Zoom an image in the PictContainer plug-in area to 100%.
- Zoom an image in the PictContainer plug-in area to 400%.
- Drag an image in the PictContainer plug-in area to a 4D Picture variable.
- Drag an image from a 4D Picture variable to the PictContainer plug-in area.
- Drag an image from a 4D List box to the PictContainer plug-in area.

Note: the 4D Pack plug-in must be installed in order to use the PictContainer plug-in on Windows.

PictContainer Plug-In Command Reference

The following section documents the commands available in the PictContainer plug-in. The commands are organized by the following themes:

- **Area Content**
- **Offscreen Areas**
- **Drag & Drop**
- **Area and Picture Properties**

Area Content

These commands deal with the content of PictContainer plug-in areas.

PC Load Picture **Area Content**

PC Load Picture(Area; picture) -> error code

Parameter	**Type**	**Description**
Area	Longint	-> PictContainer area
Picture	4D Picture	-> Picture Variable/field
Function result	Longint	<- Error code (0 = No error)

Description
The command opens an image passed as 4D (Mac) picture. On Windows the picture is converted to EMF.

PC Load File **Area Content**

PC Load File(Area; Document path) -> error code

Parameter	**Type**	**Description**
Area	Longint	-> PictContainer area
Documentpath	Text	-> Full path to document
Function result	Longint	<- Error code (0 = No error)

Description

This command opens an image from the specified path. On Windows the image is displayed from disk (the image is not read into memory to reduce memory usage), which means the file is kept open. On Mac OS the image is read into memory using QuickTime.

PC Load Blob — Area Content

PC Load Blob(Area; Blob) -> error code

Parameter	Type	Description
Area	Longint	-> PictContainer area
Blob	Blob	-> Blob containing image
Function result	Longint	<- Error code (0 = No error)

Description

The command opens an image contained inside the BLOB. The BLOB must contain an image in a supported format (like JPG). A typical way to create the BLOB is with the DOCUMENT TO BLOB command.

PC Save Picture — Area Content

PC Save Picture(Area; picture) -> error code

Parameter	Type	Description
Area	Longint	-> PictContainer area
Picture	4D Picture	<- Picture Variable
Function result	Longint	<- Error code (0 = No error)

Description

The command stores the displayed picture in a 4D Picture variable.

On Mac OS the picture is already in PICT format and is simply passed to 4D. On Windows the image is converted to BMP and then, using 4D Pack, to a Mac OS PICT. This allows the plug-in to work without QuickTime and still create Mac pictures. If 4D Pack is not installed error code -15002 is returned.

PC Save File Area Content

PC Save File(Area; Document path; Format; Compression) -> error code

Parameter	Type	Description
Area	Longint	-> PictContainer area
Documentpath	Text	-> Full path to document
Format	Text	-> Image format
Compression	Longint	-> Compression level (0-100)
Function result	Longint	<- Error code (0 = No error)

Description

The command saves an image to the specified path.

Format defines the used image format, allowed values are: jpg, tif, bmp, png and gif. On Mac OS PICT is also supported.

Compression defines the compression level, where 1 is greatest compression and 100 is least compression. Pass 0 for default compression. Compression is supported for JPG only.

Note for Windows: it is not possible to save over a file but it can be saved using another extension (format) or to be saved as BLOB or picture.

Note for Mac OS: all loaded pictures are converted to Mac OS PICT. When saving they are converted to the original type.

PC Save Blob Area Content

PC Save Blob(Area; Blob; Format; Compression) -> error code

Parameter	Type	Description
Area	Longint	-> PictContainer area
Blob	Blob	<- Picture in native format
Format	Text	-> Image format
Compression	Longint	-> Compression level (0-100)
Function result	Longint	<- Error code (0 = No error)

Description

The command stores an image as BLOB with the defined format.

Format defines the used image format, allowed values are: jpg, tif, bmp, png and gif. On Mac OS PICT is also supported.

Compression defines the compression level, where 1 is greatest compression and 100 is least compression. Pass 0 for default compression. Compression is supported for JPG only.

Note for Windows: it is not possible to save over a file but it can be saved using another extension (format) or to be saved as BLOB or picture.

Note for Mac OS: all loaded pictures are converted to Mac OS PICT. When saving they are converted to the original type.

PC Clear Picture **Area Content**

PC Clear Picture(Area)

Parameter	**Type**	**Description**
Area	Longint	-> PictContainer area

Description
The command clears the PictContainer plug-in area specified by `Area`, releasing used memory. On Windows the file locked with PC Open File is closed. The area content is cleared.

Offscreen Areas

Two commands allow using offscreen areas:

PC Create Offscreen Area **Offscreen Areas**

PC Create Offscreen Area -> Area

Parameter	**Type**	**Description**
Function result	Longint	<- PictContainer Area

Description
PC Create Offscreen Area creates a PictContainer plug-in area and returns the area’s ID. The value returned can be used in any other PictContainer command that requires an area.

It is the developer’s responsibility to close the area using the command PC Delete Offscreen Area when finished using the area to free the occupied memory.

Example
The following code shows how to open a picture offscreen and receive the picture information:

```
C_LONGINT(lAreaRef)
lAreaRef:=PC Create Offscreen Area
tPath2File := “C:\\mypicture.jpg”
lMovieFileLastError:=PC Load File (lAreaRef;tPath2File)
lMovieFileLastError:=PC Get Picture Propertes (lAreaRef; $ResX; $ResY; $Width; $Height;
$Format)
```

```
    PC Delete  Offscreen  Area  (lAreaRef)
```

PC Delete Offscreen Area **Offscreen Areas**

PC Delete Offscreen Area(Area)

Parameter	**Type**	**Description**
Area	Longint	-> PictContainer Area

Description
PC Delete Offscreen Area disposes of a PictContainer offscreen area that was created with PC Create Offscreen Area and frees the memory used.

Area and Picture Properties

PC Set Display Properties **Area and Picture Properties**

PC Set Display Properties(Area; ZoomX, ZoomY, ScrollX, ScrollY) -> error code

Parameter	**Type**	**Description**
Area	Longint	-> PictContainer area
ZoomX	Longint	-> Horizontal Zoom
ZoomY	Longint	-> Vertical Zoom
ScrollX	Longint	-> Horizontal Scroll Position
ScrollY	Longint	-> Vertical Scroll Position
Function result	Longint	<- Error code (0 = No error)

Description
PC Set Display Properties is used to modify the display properties of a PictContainer plug-in area.

`ZoomX` and `ZoomY` are used to resize the image. Use positive values to express the size in percent:100 for original size, 200 for double size, 50 for half size.
The following constants allow automatic resizing:

PC_ScaledToFitProp	0	Rescaled to fill area keeping proportion
PC_ScaledToFitProp	-1	Rescaled to fill area keeping proportion, centered
PC_ScaledToFit	-2	Rescaled to fill area
PC_ReducedToFitPropCentered	-3	Reduced (not enlarged) prop, centered
PC_ReducedToFitProp	-4	Reduced (not enlarged) proportional

`ScrollX` and `ScrollY` set the current scroll position. The value is expressed in pixels and the range is between 0 and the size of the image minus the size of the area.

PC Get Display Properties   Area and Picture Properties

PC Get Display Properties(Area; ZoomX, ZoomY, ScrollX, ScrollY) -> error code

Parameter	**Type**	**Description**
Area	Longint	-> PictContainer area
ZoomX	Longint	<- Horizontal Zoom
ZoomY	Longint	<- Vertical Zoom
ScrollX	Longint	<- Horizontal Scroll Position
ScrollY	Longint	<- Vertical Scroll Position
Function result	Longint	<- Error code (0 = No error)

Description

PC Get Display Properties returns the current display settings. See PC Set Display Properties for details.

PC Get Picture Properties   Area and Picture Properties

PC Get Picture Properties(Area; ResX; ResY; SizeX; SizeY; Format) -> error code

Parameter	**Type**	**Description**
Area	Longint	-> PictContainer area
ResX	Longint	<- Horizontal Resolution
ResY	Longint	<- Vertical Resolution
SizeX	Longint	<- Picture Width
SizeY	Longint	<- Picture Height
Format	Text	<- Picture Format
Function result	Longint	<- Error code (0 = No error)

Description

PC Get Picture Properties returns information about the current image. `ResX`, `ResY`, `SizeX`, and `SizeY` are expressed in pixels. `Format` describes the original picture format.

The possible values for `Format` on Windows are: JPEG, BMP, EMF, WMF, PNG, GIF, TIF, ICN

On Mac OS the value is set by QuickTime.

PC Get Thumbnail   Area and Picture Properties

PC Get Thumbnail(Area; Picture; Size) -> error code

Parameter	**Type**	**Description**
Area	Longint	-> PictContainer area
Picture	Picture	-> Thumbnail
Size	Longint	-> Prefered Size, Windows only

Description
PC Get Thumbnail returns a reduced version of the image as a 4D/Mac Picture.
On Mac QuickTime decides the size for the thumbnail, usually 80 pixels.

On Windows the 3rd Parameter `Size` allows the preferred size to be defined. 48 pixels is the default size. The thumbnail is created proportional. If the picture already contains a thumbnail the existing thumbnail is returned and if necessary rescaled. If the requested size is larger than the size of the existing thumbnail the picture is enlarged which can lead to a bad quality.

On Windows 4D Pack is needed. If this plug-in is not installed error -15002 is returned.

PC Set Area Properties — Area and Picture Properties

PC Set Area Properties(Area; AllowCopy, AllowPaste, AutoDrop)

Parameter	**Type**	**Description**
Area	Longint	-> PictContainer area
AllowCopy	Longint	-> Edit Menu Copy allowed
AllowPaste	Longint	-> Edit Menu Cut/Clear allowed
AutoDrop	Longint	-> Drop of pictures allowed

Description
PC Set Area Properties allows some features to be disabled.

By default all features are enabled. If the plug-in area contains an image, the user can use the Edit menu command "Copy" to copy the image into the. Passing a 0 for parameter `AllowCopy` disables the copy menu item. The same may be done for the parameter `AllowPaste`. If set to 0 the Paste, Clear and Cut commands are always disabled, preventing the user from modifying the displayed image using the Edit menu. As a result the Cut menu item is only enabled if the area contains an image and both copy and paste are allowed (to put a copy into the clipboard and to modify the area content to clear it).

The parameter `AutoDrop` disables the automatic drop support for picture objects. Drop can still be handled using the programming language. See chapter Drag & Drop for more details.

PC Get modified **Area and Picture Properties**

PC Get modified(Area) -> modified

Parameter	Type	Description
Area	Longint	-> PictContainer area
Function result	Longint	<- 0 = unmodified / 1 = modified

Description

PC Get modified allows checks to see if the user modified the displayed picture, either by using the Edit menu or using Drag & Drop. Modifying the picture using the language does not return 1.

Drag & Drop

If the “drag enabled” check box is enabled for a PictContainer plug-in area, the plug-in automatically allows the dragging of the picture to other objects. The developer needs to handle the 4D events, e.g. if DRAG AND DROP PROPERTIES returns the plug-in object then the current image can be received using PC Save Picture, then assigned to another 4D object.

The plug-in handles drag operations from picture variables, fields and scrollable areas from the current form (same process as the plug-in) automatically without the need to use the language. The developer needs to enable the “drop enabled” check box in the object properties for the plug-in to activate this feature.

A drag operation from another process is more complicated. For variables the developer can receive the content of a variable from another process using GET PROCESS VARIABLE, but for fields it is more complicated, because it depends on the current record of the other process. Usually the developer uses a concept like CALL PROCESS. Even a drag from the same process is more complicated for objects like a list box, which may contain several columns or other objects defining the picture, but not containing it. To support these concepts the plug-in allows the developer to control the drag operation manually through the programming language, using the command PC Allow Drop. It is recommended that all drags be handled manually, even standard pictures, to have a common way in the code. In this case the automatic drag support can be disabled using PC SET AREA PROPERTIES.

PC Set Allow Drop **Drag & Drop**

PC Set Allow Drop(Area)

Parameter	Type	Description
Area	Longint	-> PictContainer area

Description
PC Set Allow Drop enables the drop acceptance for the current drop operation. This command must be called from the On Drag Over form event for the plug-in area object method.

The plug-in normally accepts drops automatically from picture variables, fields and scrollable areas from the current process (except if disabled using PC Set Area Properties). This command makes it possible to handle other processes or other objects, like List box objects or other plug-ins.

If this command is used during On Drag Over the plug-in accepts the drag over assuming if the user releases the mouse on top of the plug-in area and the object method is called with an On Drop event. The developer can handle the drop using standard 4D drag & drop commands.

Example
Suppose a form contains a PictContainer area named “pictarea” and a list box named “list box1”, including a picture column named “column1”. To allow drag & drop of the list box to the PictContainer area, the object method might look like this:

```
Case of
   : (Form event=On Drag Over )
      DRAG AND DROP PROPERTIES($source;$sourceelement;$sourceprocess)
      If ($source=(->List box1))
         PC Set Allow Drop (pictarea;1)
      End if
   : (Form event=On Drop )
      DRAG AND DROP PROPERTIES($source;$sourceelement;$sourceprocess)
      If ($source=(->List box1))
         $err:=PC Load Picture (pictarea;column1{$sourceelement})
      End if
End case
```

Conclusion

The PictContainer plug-in provides enhanced support for picture handling with 4D. This plug-in exposes functionality that is natively available on both Windows and Mac OS computers so as to reduce installation requirements.

GDI+ for Windows 2000

While GDI+ is preinstalled on Windows XP, Windows 2003 and future systems, it was not part of Windows 2000. Chances are good that the DLL is already existing, but it is preferred that it be shipped with the application. Installation is simple, put it beside 4D.exe (or 4DClient.exe, etc). The redistributable DLL can be downloaded here:

http://www.microsoft.com/msdownload/platformsdk/sdkupdate/psdkredist.htm

PictContainer Error codes

Some commands may return special errors:

15001	Unknown/invalid area ID
15002	4D Pack not installed (needed on Windows to create 4D Picture)
15003	Thumbnail creation error (Windows only)

Windows Error Codes

On Windows the commands may return the following (positive) error codes from GDI+:

1	GenericError
2	InvalidParameter
3	OutOfMemory
4	ObjectBusy
5	InsufficientBuffer
6	NotImplemented
7	Win32Error
8	WrongState
9	Aborted
10	FileNotFound
11	ValueOverflow
12	AccessDenied
13	UnknownImageFormat
14	FontFamilyNotFound
15	FontStyleNotFound
16	NotTrueTypeFont
17	UnsupportedGdiplusVersion
18	GdiplusNotInitialized
19	PropertyNotFound
20	PropertyNotSupported
21	ProfileNotFound

Note: Load and Save often return OutOfMemory, even there is plenty of memory, for example: invalid path name. This is because several GDI+ constructors set the status to OutOfMemory when they fail regardless of the reason for failure.

Mac OS Error Codes

On Mac OS returned error codes are standard QuickTime or Mac OS error codes, as listed in 4D documentation.