

Picture Management in 4D v11 SQL

By Luis Pineiros Technical Services Team Member, 4D Inc.

Technical Note TN 08-31

Table of Contents

Table of Contents	2
Abstract	3
Introduction	3
Pictures Overview	3
Native Support	3
Drag and Drop.....	4
Contextual Menus	6
Scrollbars.....	7
Picture Formats	8
Picture Codec IDs	8
Picture Conversion	9
Picture Storage.....	9
Picture Fields.....	9
Resources folder	10
Picture Commands.....	13
TRANSFORM PICTURE	13
SCALE.....	14
FLIP HORIZONTALLY	15
FLIP VERTICALLY	16
CROP	17
FADE TO GREY SCALE	18
COMBINE PICTURES.....	19
Conclusion	24

Abstract

Picture Management in 4D v11 SQL has been enhanced to offer native support of Mac OS X and Windows picture formats. It is now possible to combine pictures without losing their original characteristics, as well as apply much more sophisticated transformations on a picture.

This Technical Note explores this new functionality. A sample database is provided.

Introduction

4D v11 SQL offers native support of current picture formats available in Mac OS X and Windows. Without having to rely on interpretation, the specific features of these different formats, such as transparency, shading, color, etc., can be used and retained in 4D without alteration.

Picture operations now offer great flexibility to concatenate or superimpose images to create combined pictures that retain their original characteristics.

Local coordinates of a click on a picture field or variable can be retrieved, making it possible to have precise control of an image.

The process of transforming a picture now provides a powerful set of operations. You can scale, move, flip horizontally and vertically, crop and fade an image to grey scale.

Pictures Overview

Picture Management in 4D v11 SQL has been prioritized to provide native support. Therefore, Picture fields and variables are now identical as far as interface and functionality.

Picture variables in forms must be typed in the Property List or before loading the form; that is even before the On Load form event.

Two possibilities are either a C_PICTURE (pict_var_name) statement before loading the form, or typing the variable in the **Variable Type** menu of the Property List. Unless you use one of these two methods to declare the variable, 4D v11 SQL will not display the variable correctly in Interpreted mode.

Native Support

Pictures are now stored in 4D v11 SQL in their original format, without any interpretation in 4D. This means that specific features of the original format such as shading, transparency, etc. will be retained and displayed without any alteration. One such format is an animated GIF, which in order to “play”, needs to be displayed as a static picture or a library picture. It is not possible to “play” an animated GIF stored in a picture variable or field.

The native picture support applies to all pictures stored in 4D v11 SQL, including library pictures, pictures pasted into forms, pictures in variables or fields and pictures in the Resources folder. Furthermore, pictures retain their characteristics whether they are pasted, imported using a contextual menu, imported programmatically, exported or dragged and dropped.

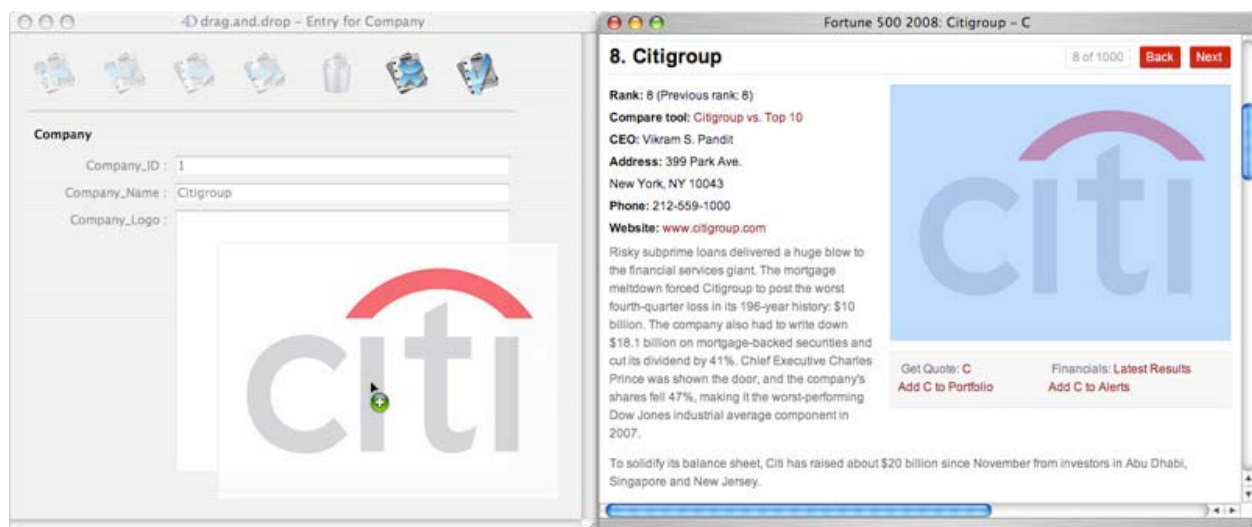
In case 4D cannot display a picture format, it calls on QuickTime routines.

Drag and Drop

4D v11 SQL automatically manages drag and drop between objects, both fields and variables.

4D v11 SQL also manages drag and drop between 4D and a third-party application containing a picture or a list of files from the Operating System. It works by simply dragging a picture from the application and dropping it into 4D. You can drop the picture into a picture library, a variable or a field.

This is an example of a picture in a Web site being dragged and dropped into a picture field in a form.



You can control whether to allow drag and drop from the Action theme of the Property List of the object.

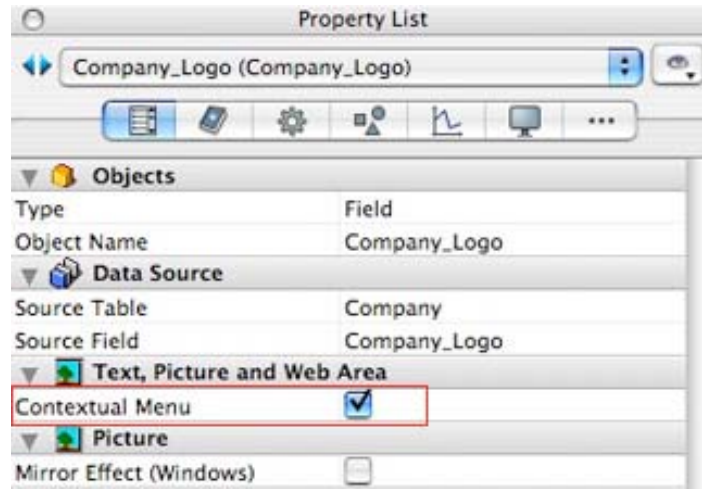
Draggable and **Droppable** apply to automatic Drag and Drop options between 4D v11 SQL objects. **Automatic Drop** refers to the option between another application in your system and 4D v11 SQL.



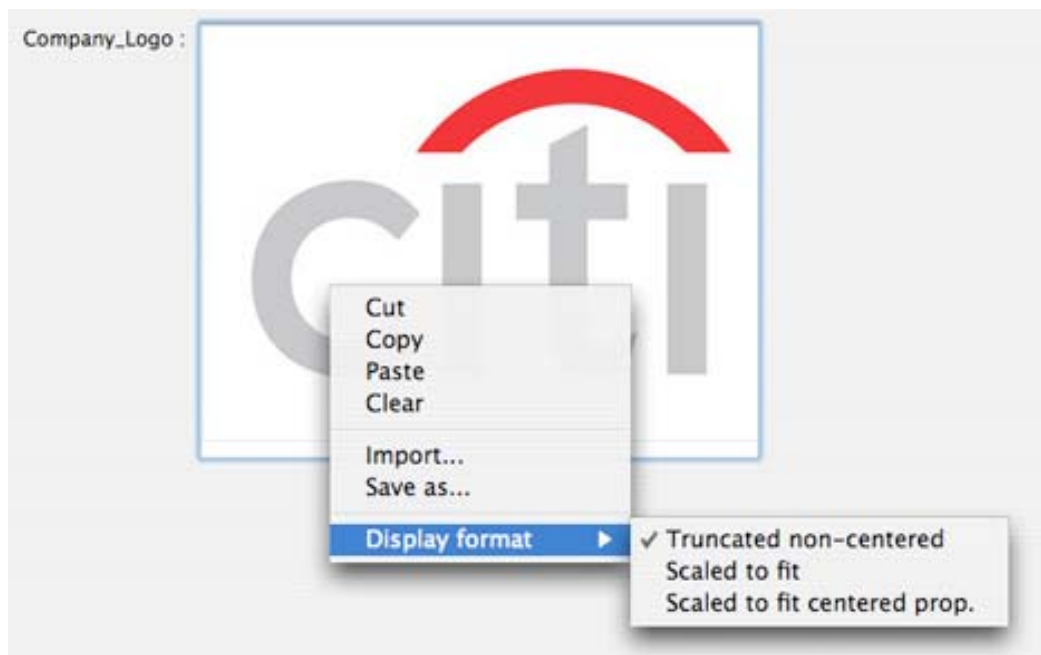
Contextual Menus

4D v11 SQL Release 2 (11.2) also provides picture fields and variables with a contextual menu.

Check the **Contextual Menu** option in the Text and Picture theme of the Property List.



With the Contextual Menu activated, you can access the menu by Control-Clicking on Mac OS X or the right mouse button on Windows.



The displayed menu offers the standard OS Cut, Copy, Paste and Clear options. In addition, it allows you to **Import** a picture to the field from a file or **Save** the picture to a file. These two commands use the native picture management in 4D v11.2 SQL and can be used with any of the native formats supported by 4D.

The menu can also be used to modify the display format of the picture. Three different formats are offered: **Truncated non-centered**, **Scaled to fit** and **Scaled to fit centered prop**. Keep in mind that these modifications are only used temporarily to display the picture; they are not saved with the record.

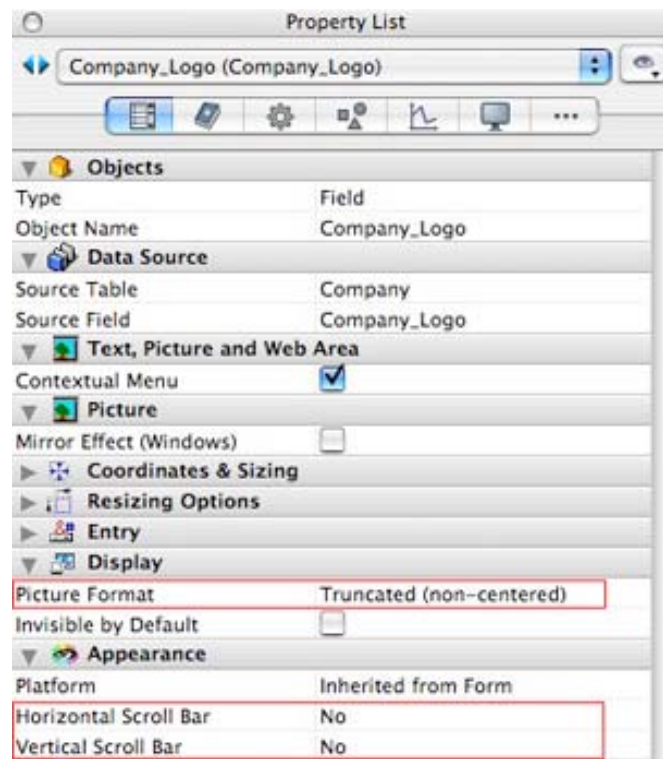
If the field or variable are not assigned as enterable, only the **Copy**, **Save As...** and the **Display format** commands are available.

Scrollbars

Picture type objects can have scrollbars. To activate scrollbars for pictures, the display format must be set to **Truncated (non-centered)**.

The **Horizontal Scroll Bar** and **Vertical Scroll Bar** properties will become active in the Appearance theme of the Property List. Each property has three options:

- **Yes**: The scrollbar is always visible. (Even when it is not necessary)
- **No**: The scrollbar is never visible.
- **Automatic**: The scrollbar appears only when necessary.



You can also use the **SET SCROLLBAR VISIBLE** command to set scrollbars.

Picture Formats

The native formats supported by your OS are available in 4D v11 SQL Release 2 (11.2). The **PICTURE CODEC LIST** command can be used to obtain a list of the formats in your Operating System and configuration.

For more detailed information about the **PICTURE CODEC LIST** command:

[PICTURE CODEC LIST](#)

These are the supported formats:

4D picture (.4pct)
JPEG image (.jpg .jif .jpeg .jpe)
Portable Network Graphics image (.pgn)
Paint Brush image (.bmp .dib .rle)
Graphics Interchange Format (.gif)
Radiance (Mac OS X Only)
Portable Document Format (Mac OS X Only) (.pdf)
TIFF image (.tif .tiff)
Paint Brush image (Windows Only) (.emf)
PICT image (.pic .pct .pict)
Safari document (.svg)
Silicon Graphics image (.sgi .rgb)
Adobe Photoshop image (.psd)
Mac Paint image (.pntg .pnt .mac)
Truevision TGA image (.targa .tga)
JPEG 2000 image (.jp2)
QuickTime image (.qtif .qti)

Picture Codec IDs

Picture formats compatible with 4D v11 SQL Release 2 (11.2) are returned by the **PICTURE CODEC LIST** command as picture Codec IDs in three different forms:

- Extension (ex. .gif)
- Mime Type (ex. image/jpeg)
- 4-character QuickTime code (ex. PNTG)

The form returned depends on the way the Codec is recorded at the OS level. Most picture management commands can receive a Codec ID as a parameter. Therefore, you should always use the system ID returned by the **PICTURE CODEC LIST** command.

Picture Conversion

4D v11 SQL Release 2 (11.2) allows you to convert a picture from one format to another. All the formats supported are available in the conversion. You can use the **CONVERT PICTURE** command to perform the conversion.

For more detailed information about the **CONVERT PICTURE** command:

[CONVERT PICTURE](#)

Picture Storage

There are a few different ways of storing pictures in 4D v11 SQL. Pictures can be saved in fields, stored in the Pictures Library, placed as static image objects in forms, temporarily displayed in picture variables or stored in the Resources folder.

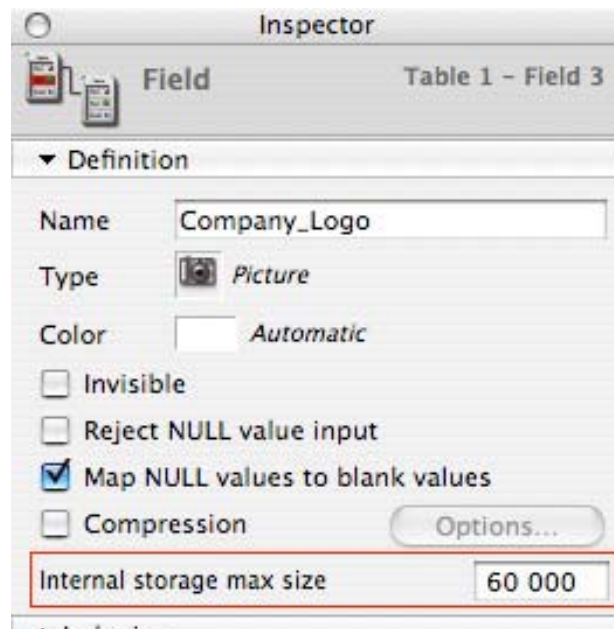
As discussed previously, all these different forms of picture storage and management are now handled by 4D with native support for most of the current picture formats available in your Operating System. This means that regardless of the method you utilize to manage pictures in your database, 4D v11.2 SQL will retain the original characteristics of the format used in the picture. This is very important considering that in previous versions of 4D, pictures were always converted before being managed by 4D, in the process losing their original format options such as transparency, shading, etc.

Picture Fields

Picture fields allow you to have a picture as a part of the record.

In versions of 4D v11 SQL previous to v11.2, picture fields were stored outside of records for optimization purposes. However, in 4D v11 SQL Release 2 (11.2) you can choose when picture fields get stored outside of a record. Depending on the size of the pictures being stored in your database, you may want to consider storing them in the record, especially if most of the pictures are of limited size.

In order to set the size that determines when to store the picture in the record, you need to select the field and display the Inspector window in the Database Structure:



In this case, a value of 60 000 means approximately 60K; therefore, a picture with less than 60K will be stored in the record and a picture with more than 60K outside the record. By default, 4D assigns 0 to this value; this means all pictures will be stored outside of the record.

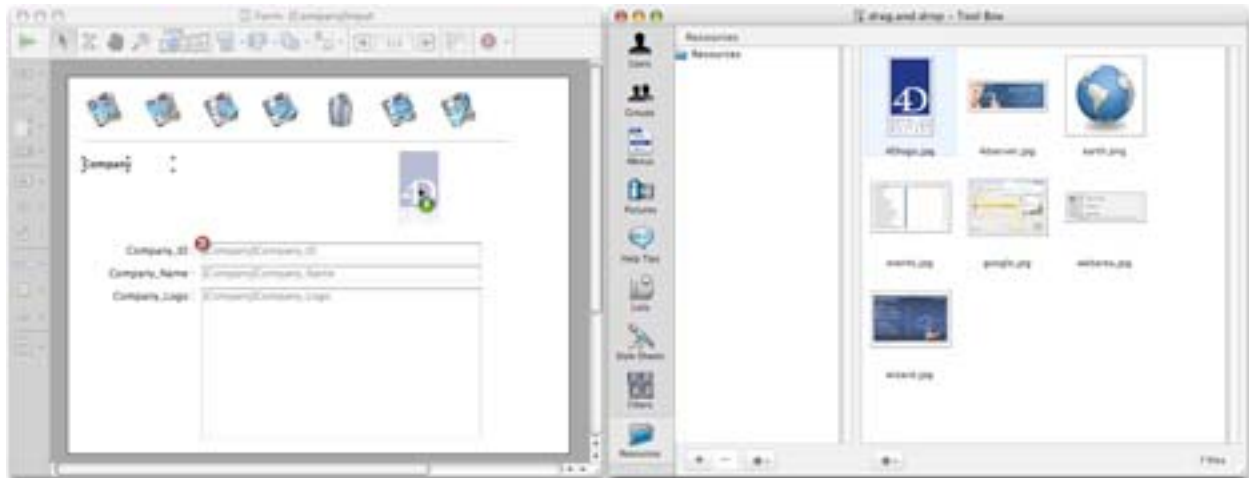
Resources folder

The Resources folder can contain external resources of various types. Picture files, text files, XLIFF files, etc. Our main interest for the purpose of this Technical Note is how the Resources folder handles pictures.

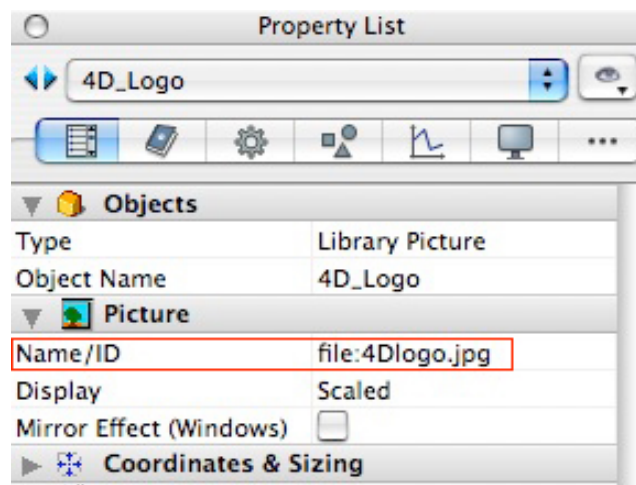
The Resources folder allows you to store picture files that you can use in your forms or records by using a reference of the file. The display of the picture is optimized and managing them is more efficient.

To make the managing of the contents of the Resource folder easier, 4D v11 SQL introduced the Resources Tab on the Tool Box. To place a picture in a form, simply drag and drop the picture from the Resources manager. 4D v11 SQL will automatically reference the object with a **file:{pathname+}filename** Name/ID.

This is an example of how to drag and drop into a form:



This is the result from the Picture's Property List information.



You can also access pictures stored in the Resources folder programmatically. This is an example of code to accomplish that:

```
C_PICTURE($0)
C_TEXT($1)

C_PICTURE($Pict_Res)
C_TEXT($Resource_Path;$Separator)
C_INTEGER($Platform)

$Resource_Path:=Get 4D folder(Current Resources folder )

PLATFORM PROPERTIES($Platform)

If ($Platform=Windows )
    $separator:="\\"
Else `Mac OS
    $Separator:=":"
End if
```

```
$Resource_Path:=$Resource_Path+$1
$Resource_Path:=Replace string($Resource_Path;"/";$Separator)

If (Test path name($Resource_Path)=Is a document )
    READ PICTURE FILE($Resource_Path;$Pict_Res;*)
    If (OK=1)
        $0:=$Pict_Res
    End if
Else
    ALERT("Document does not exist...")
End if
```



SCALE

We can scale a picture up or down on its width, height or both.

``Scales the picture 75% proportionally on the width and height.`

```
TRANSFORM PICTURE(var_pict_4D_map;Scale ;0.75;0.75)
```



FLIP HORIZONTALLY

We can flip a picture horizontally.

```
TRANSFORM PICTURE(var_pict_4D_map;Flip horizontally)
```



FLIP VERTICALLY

We can flip a picture vertically.

```
TRANSFORM PICTURE(var_pict_4D_map;Flip vertically)
```



CROP

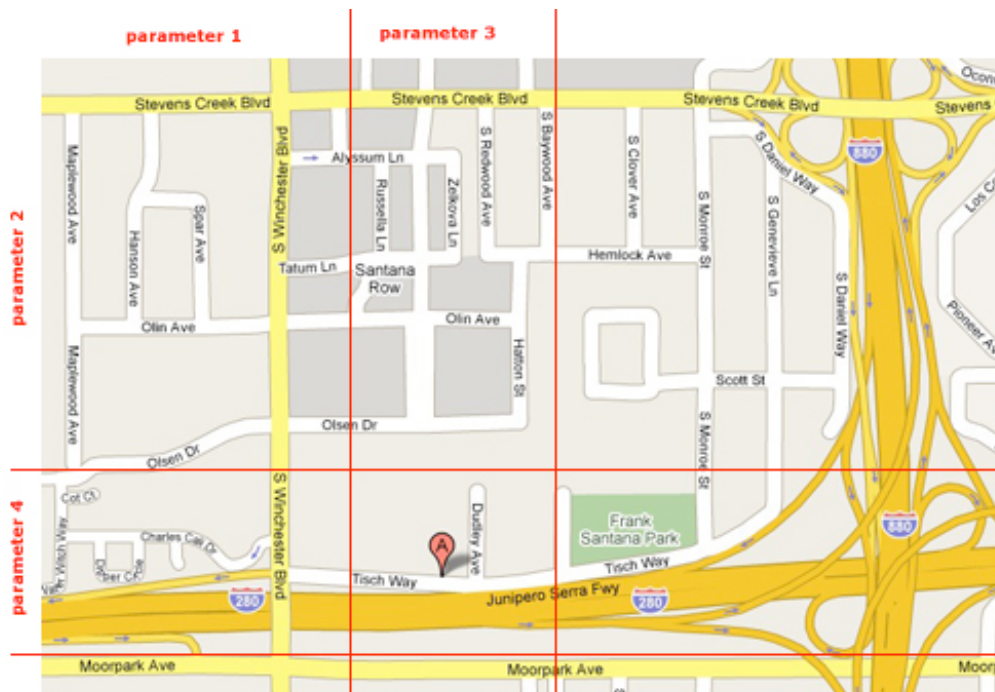
We can crop a picture with very precise dimensions. For instance, if we wanted to concentrate just on the area within the address label, we would have to designate the four parameters for the cropping action. Parameter 1 determines the X originating position. Parameter 2 determines the Y originating position. Parameter 3 determines the Width of the crop. Parameter 4 determines the Height of the crop. All these measurements are in pixels.

```
TRANSFORM PICTURE(var_pict_4D_map;crop;150;200;100;90)
```

This is the result:



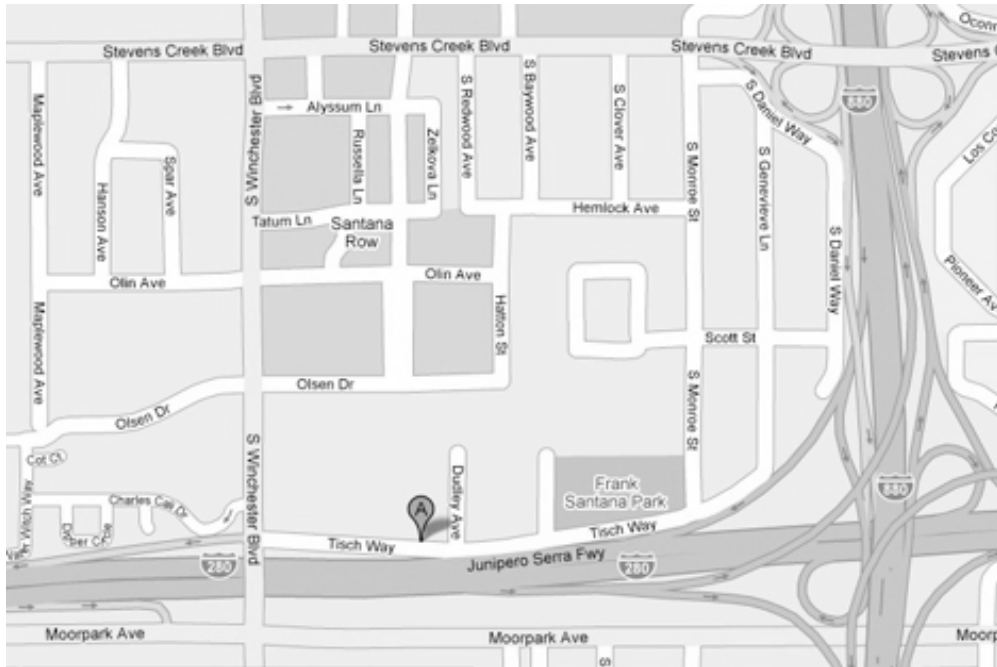
Here's how the image was cropped:



FADE TO GREY SCALE

Turns the picture into a grey scale picture. It cannot be undone.

```
TRANSFORM PICTURE(var_pict_4D_map;Fade to grey scale)
```



COMBINE PICTURES

The **COMBINE PICTURES** command can be used to obtain a combined version of two pictures by applying a combination operator and parameters.

For more information about the COMBINE PICTURES command refer to the documentation:

[COMBINE PICTURES](#)

Lets explore the possibilities that this command offers:

These are the original images used to make the combinations. The map is a **jpg** image and the logo is a **png** image with transparency (a border has been added for emphasis).



`Combine the pictures with horizontal concatenation

```
COMBINE PICTURES(var_pict_4D_map_combined;var_pict_4D_map;Horizontal  
concatenation;var_pict_4D_logo)
```



`Combine the pictures with vertical concatenation

```
COMBINE PICTURES(var_pict_4D_map_combined;var_pict_4D_map;Vertical  
concatenation;var_pict_4D_logo)
```



`Combine the pictures with superimposition

```
COMBINE PICTURES(var_pict_4D_map_combined;var_pict_4D_map;Superimposition  
;var_pict_4D_logo;100;90)
```

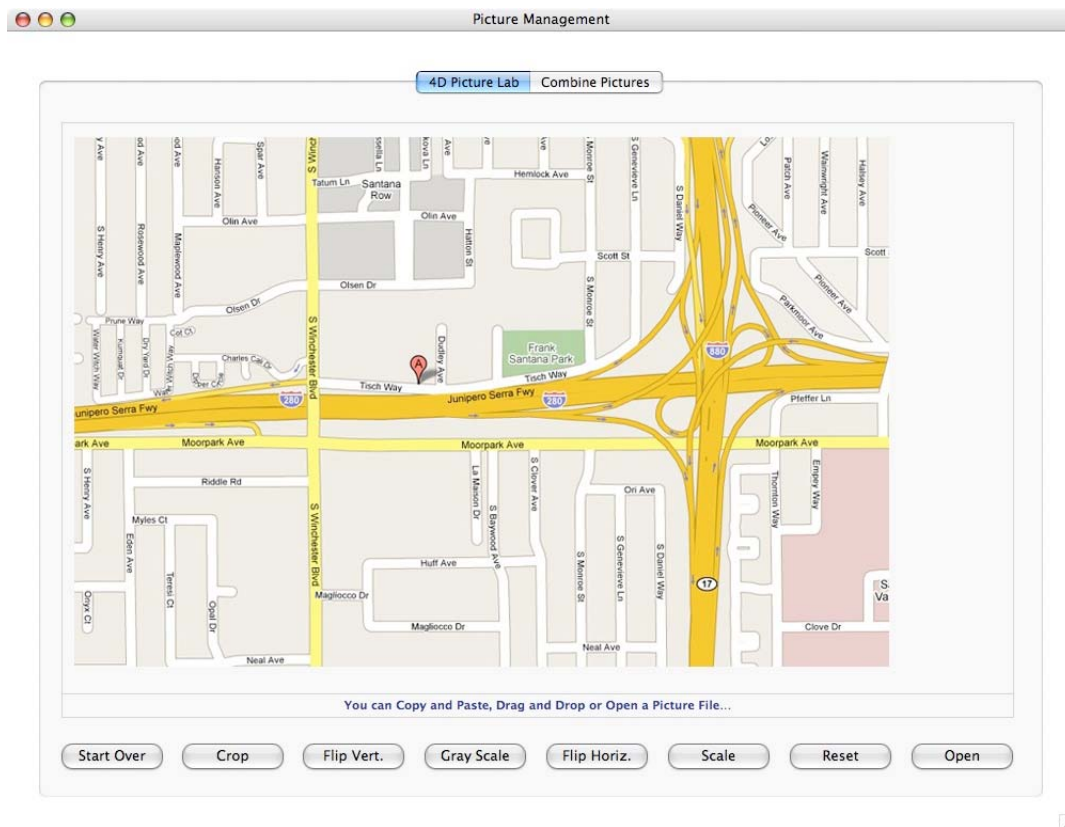


Sample Database Description

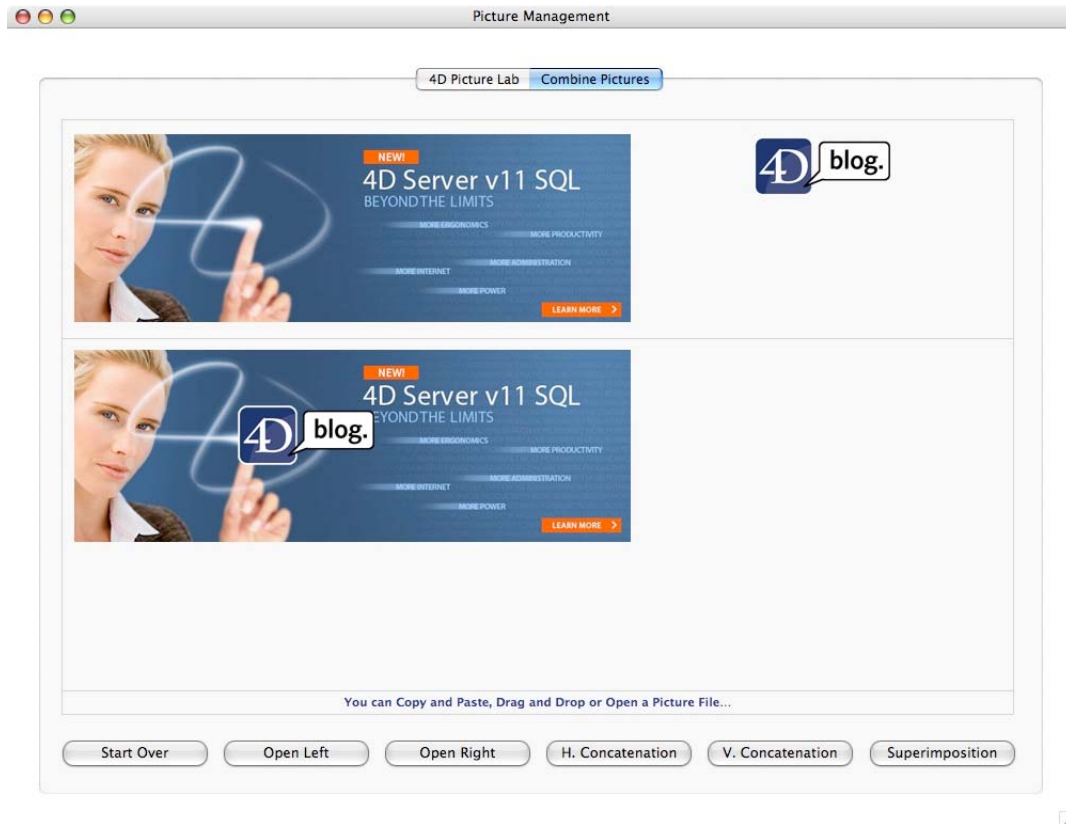
The 4D Picture Management database demonstrates the new commands and picture management features of 4D v11 SQL.

The splashscreen contains an animated GIF to showcase the ability 4D v11 SQL has to maintain the format of an image file in a layout without losing its original characteristics.

The demo is a 2-page form with a variable that gets an image loaded from the Resources folder. The first page contains the **4D Picture Lab** ; this section allows you to try the Crop, Flip Vertical, Flip Horizontal, Gray Scale and Scale functions.



The second page has the **Combine Pictures** section, it allows you to combine images by Horizontal Concatenation, Vertical Concatenation or Superimposition. In both sections you can Open a picture file(s), Copy and Paste or Drag and Drop an image ; as well as reload the original image(s) by clicking on Start Over.



The 4D Picture Management database uses the following Project Methods:

4D_PM_CombinePict:

Combines two images using different Operators.

4D_PM_LoadPict:

Loads an image from the Resources folder.

4D_PM_OpenPict:

Opens a picture file.

4D_PM_Quit:

Quits the application with a confirm dialog.

4D_PM_StartDemo:

Starts a new process for the Demo.

4D_PM_StartDemo_New:

Opens the 4D Picture Management Demo screen

4D_PM _Utility_AboutDemo:

About Dialog.

4D_PM _Utility_CenterWindow:

Opens a window centered on the screen.

4D_PM _Utility_CloseWindow:

Closes window when close box is clicked.

Conclusion

4D v11 SQL offers an extraordinary set of tools and technology to manage pictures. You can take advantage of these powerful new features in your development as well as data management. Pictures are now stored and managed by 4D v11 SQL in native format, with no alterations to the original characteristics of the images. 4D no longer transforms images to PICT and it can work with almost all the formats available in your Operating System.

In addition, two important optimizations have also been implemented. The first is to have pictures selected in forms enclosed by a selection rectangle (focus), just like any other form object. In previous versions of 4D, pictures were displayed in reverse video. Second, when the same picture (field or variable) is used in several locations in the same form, only one occurrence of the picture is now created in memory.

Now is the perfect time to give your interface a second look and consider implementing more graphic-intensive elements; 4D v11 SQL not only will be able to keep up with it, but the tools at your disposal are much more powerful. You can also make 4D v11 SQL the engine to manage databases that can handle any number of images in many different formats, without sacrificing the formatting characteristics of the images, or performance.