# The DBGridEh.LoadAndRenderHeavyImages project demonstrates the capabilities of DBGridEh and MemTableEh for loading, storing and displaying large graphic files.

There are 4 implementations for working with graphic files in the project.

**Demo1 - Simple implementation.**

A table with links to pictures is downloaded from www.ehlib.com, from the FishHavyImagesDataTable.Dfm file.

The link to the picture is in the “Image” field of the FishHavyImagesDataTable.

Click the 'Load One Image' button to load one image.

If you try to scroll through the grid, you can see that the scrolling is very slow.

This happens because, firstly, the file is large, more than 1 MB, and secondly, the file is stored in the Jpg format and rendered directly from the same format.

**Demo2 - Bmp buffered implementation.**

A table with links to pictures is downloaded from www.ehlib.com, from the FishHavyImagesDataTable.Dfm file.

The link to the picture is in the “Image” field of the FishHavyImagesDataTable.

After uploading, the file is immediately converted to Bmp format and at the same time reduced in size to 400x400.

In the grid, a picture is displayed from a field that stores a thumbnail image in Bmp format.

Click the 'Load One Image' button to load one image.

If you try to scroll through the grid, you can see that the scrolling is fast. Rendering optimization is achieved by converting Jpg to Bmp format. By reducing the size, we achieve a decrease in the size of the used memory.

If you press the ‘Load All Images’ button, then within a certain time all the images will be loaded (there are no more than 25 of them). At the same time, during the download, the interactive work of the program will be frozen.

**Demo3 - Implementation with buffering and loading images in a separate thread.**

A table with links to pictures is downloaded from www.ehlib.com, from the FishHavyImagesDataTable.Dfm file.

The link to the picture is in the “Image” field of the FishHavyImagesDataTable.

After downloading, the file is immediately converted to Bmp format by analogy with the implementation in Demo2.

If you press the ‘Load All Images’ button, then within a certain time all the images will be loaded (there are no more than 25 of them). At the same time, during loading, you can continue to work with the program and scroll through the contents of the grid.

File loading is implemented using the TLoadingQueue, TLoadingQueueItem, TLadingThread classes.

The TLoadingQueue class adds information about the loading of the next image to the downloading queue. At the same time, it simultaneously takes the first item from the queue to start a separate load thread using the TLadingThread class. TLoadingQueue ensures that no more than one thread is used to load images from the queue.

**Demo4 - Implementation with buffering and loading images in a separate stream. Loading images as needed.**

A table with links to pictures is downloaded from www.ehlib.com, from the FishHavyImagesDataTable.Dfm file.

The link to the picture is in the “Image” field of the FishHavyImagesDataTable.

After downloading, the file is immediately converted to Bmp format by analogy with the implementation in Demo2.

Images are loaded in a separate thread, similar to the implementation in Demo3.

The images are loaded dynamically when the graphics cell is first drawn.