HOWTO: Read and write images from a database using SQL Server

Atalasoft is an imaging SDK and we do not directly have any dependencies on, or specific functionality for Database access (with the single exception of our DbImageSource class).

However, since the typical means of image storage in a database is a BLOB (Binary Long Object), developers can take advantage of the AtalaImage.ToByteArray and AtalaImage.FromByteArray methods to use in their own code to store and retrieve image data to and from databases respectively.

Atalasoft DotImage has streaming capabilities that can be used jointly with ADO.NET to read and write images directly to a database without saving to a temporary file. The following code snippets demonstrates this in C# and VB.NET.

Please note that the code examples below are provided as a courtesy/convenience only and are not meant to be production code or represent the only way to access a database. The code is provided as is as but an example of how to get the needed data in a byte array suitable for use in storing to a database and how to take binary data containing an image back into an AtalaImage to use in our SDK.

Write to a Database

C#

```
rivate void SaveToSqlDatabase(AtalaImage image) { SqlConnection myConnection = null; try { // Save image to byte array. // we will be storing this image as a Jpeg using our JpegEncoder with 75 quality // you could use any of our ImageEncoder classes to store as the respective image types byte[] imagedata = image.ToByteArray(new Atalasoft.Imaging.Codec.JpegEncoder(75)); // Create the SQL statement to add the image data. myConnection = new SqlConnection(CONNECTION_STRING); SqlCommand myCommand = new SqlCommand("INSERT INTO Atalasoft_Image_Database (Caption, ImageData) VALUES ('" + txtCaption.Text + "', @Image)", myConnection); SqlParameter myParameter = new SqlParameter("@Image", SqlDbType.Image, imagedata.Length); myParameter.Value = imagedata; myCommand.Parameters.Add(myParameter); // Open the connection and execute the statement. myConnection.Open(); myCommand.ExecuteNonQuery(); } finally { myConnection.Close(); } }
```

Visual Basic.NET

```
rivate Sub SaveToSqlDatabase(ByVal image As AtalaImage) Dim myConnection As SqlConnection = Nothing Try ' Save image to byte array. ' we will be storing this image as a Jpeg using our JpegEncoder with 75 quality ' you could use any of our ImageEncoder classes to store as the respective image types Dim imagedata() As Byte = image.ToByteArray(New
```

```
Atalasoft.Imaging.Codec.JpegEncoder(75)) ' Create the SQL statement to add the image data. myConnection = New SqlConnection(CONNECTION_STRING) Dim myCommand As SqlCommand = New SqlCommand("INSERT INTO Atalasoft_Image_Database (Caption, ImageData) _ VALUES ('" + txtCaption.Text + "', @Image)", myConnection) Dim myParameter As SqlParameter = New SqlParameter("@Image", SqlDbType.Image, imagedata.Length) myParameter.Value = imagedata myCommand.Parameters.Add(myParameter ' Open the connection and execute the statement. myConnection.Open() myCommand.ExecuteNonQuery() Finally myConnection.Close() End Try End Sub
```

Read from a Database

C#

```
rivate AtalaImage OpenFromSqlDatabase() { SqlConnection myConnection = null; try { // Establish connection and SELECT statement. myConnection = new SqlConnection(CONNECTION_STRING); SqlCommand myCommand = new SqlCommand("SELECT ImageData FROM Atalasoft_Image_Database WHERE Caption = '" + txtCaption.Text + "'", myConnection); myConnection.Open(); // Get the image from the database. byte[] imagedata = (byte[])myCommand.ExecuteScalar(); if (imagedata != null) { // This one line of code uses Atalasoft's RegisteredDecoders.Decoders collection // to detect the type of image and decode it back into an AtalaImage // NOTE: if the image is a multipage format, this would only provide the first frame // for multipage image support you'd need to make a MemoryStream of the imageData and then use // new AtalaImage(memoryStreamHere, frameIndexHere, null) instead of the AtalaImage.FromByteArray AtalaImage image = AtalaImage.FromByteArray(imagedata); return image; } else { MessageBox.Show("Image does not exist in database."); return null; } } finally { myConnection.Close(); } }
```

Visual Basic .NET

```
rivate Function OpenFromSqlDatabase() As AtalaImage Dim myConnection As SqlConnection = Nothing Try ' Establish connection and SELECT statement. myConnection = New SqlConnection(CONNECTION_STRING) Dim myCommand As SqlCommand = New SqlCommand("SELECT ImageData FROM Atalasoft_Image_Database _ WHERE Caption = '" + txtCaption.Text + "'", myConnection) myConnection.Open() ' Get the image from the database. Dim imagedata() As Byte = CType(myCommand.ExecuteScalar(), Byte()) If (Not imagedata Is Nothing) Then ' This one line of code uses Atalasoft's RegisteredDecoders.Decoders collection ' to detect the type of image and decode it back into an AtalaImage ' NOTE: if the image is a multi-page format, this would only provide the first frame ' for multi-page image support you'd need to make a MemoryStream of the imageData and then use ' New AtalaImage(memoryStreamHere, frameIndexHere, null) instead of the AtalaImage.FromByteArray Dim image As AtalaImage = AtalaImage.FromByteArray(imagedata) Return image Else MessageBox.Show("Image does not exist in database.") Return Nothing End If Finally myConnection.Close() End Try End Function
```

Original Article

Q10018 - HOWTO: Read and write images from a database using SQL Server

Last Update

2026-11-02 - TD - (Code newlines fixed)

Atalasoft Knowledge Base
https://www.atalasoft.com/kb2/KB/50368/HOWTO-Read-and-write-images-from-a-d...