

EasyToFind.us
ASP Store Locator

DOCUMENTATION AND
INSTALLATION PROCEDURES

Version 3.0



Reflekt Media Inc

Copyright 2002

TABLE OF CONTENTS

END USERS LICENSE AGREEMENT (EULA)	3
INSTALLATION INSTRUCTIONS	5
COM COMPONENT	7
Properties	7
Methods	8
TROUBLE SHOOTING	10
UPGRADING TO SQL SERVER	11
FAQ	12

End Users License Agreement (EULA)

ASP STORE LOCATOR (1) ONE-WEB SITE LICENSE AGREEMENT

By installing the ASP Store Locator Software you are agreeing to the following terms and conditions.

IMPORTANT-READ CAREFULLY:

This License Agreement is a legal agreement between you (either an individual or a single entity) and Reflekt Media Inc. ("Reflekt Media" or "we") for the Reflekt Media developed ASP Store Locator software (hereafter referred to as the SOFTWARE PRODUCT) accompanying this License Agreement, which includes ZIP Code distance routines and data file(s). By exercising your rights to make and use copies of the SOFTWARE PRODUCT, you agree to be bound by the terms of this License Agreement. If you do not agree to the terms of this License Agreement, you may not use the SOFTWARE PRODUCT.

GRANT OF LICENSE

This License Agreement grants the following rights: You are granted the right to include the SOFTWARE PRODUCT files on one Internet or LAN Server with one web site only. You may not use the SOFTWARE PRODUCT files on multiple computers or multiple websites on the same computer. You may make backup copies for archival purposes.

DESCRIPTION OF LIMITATIONS

You may not reverse engineer, decompile, or disassemble the file(s), except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation. Without prejudice to any other rights, Reflekt Media may terminate this License Agreement if you fail to comply with the terms and conditions of this License Agreement. In such event, you must destroy all copies of the SOFTWARE PRODUCT and all of its component parts.

COPYRIGHT

All title and copyrights in and to the SOFTWARE PRODUCT and any copies of the SOFTWARE PRODUCT are owned by Reflekt Media. The SOFTWARE PRODUCT is protected by copyright laws and international treaty provisions.

NO WARRANTIES

Reflekt Media expressly disclaims any warranty for the SOFTWARE PRODUCT. The SOFTWARE PRODUCT and any related documentation is provided "as is" without warranty of any kind, either express or implied, including, without limitation, the implied warranties of merchantability or fitness for a particular purpose. The entire

risk arising out of use or performance of the SOFTWARE PRODUCT remains with you.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES

In no event shall Reflekt Media nor anyone else who has been involved in the creation, production, or delivery of the SOFTWARE PRODUCT be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss) arising out of the use of or inability to use this Reflekt Media product, even if Reflekt Media has been advised of the possibility of such damages. Because some states and jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

MISCELLANEOUS

If you acquired this product in the United States, this License Agreement is governed by the laws of the State of Washington.

Reflekt Media Inc.
7541 Mary Avenue Northwest
Seattle, WA 998117
(253) 226-6455
<http://www.reflektmedia.com>

Installation Instructions

The ASP Store Locator requires a few steps to get up and running properly. The following instructions are the most thorough installation procedure you can follow to insure correct installation.

- 1) Unzip the entire contents of the zip file into a temporary folder location of your choice.
- 2) Install the DLL and Windows testing application by double clicking on the setup.exe found in the "Application" folder and follow the setup wizard.
- 3) Move the Access database found in the Database folder to the final destination of your choice. Note : Do not install the database inside your wwwroot folder because people could download your database if they know the correct path.
- 4) Next create a database connection string by opening up the UDL folder and double clicking on the connection.udl file. NOTE : If you know how to create a connection string then you may skip to Step 6
 - Click on the *Provider* tab and choose the following provider.
For Access select : **Microsoft Jet OLEDB 4.0**
 - Click on the *Connection* tab and click the ... button and navigate to the Access Database you relocated in step 3.
 - Click on the *TEST CONNECTION* button and validate your connection information is correct. Do not continue with installation until you get a successful confirmation. Click OK
- 5) Start instances of Notepad and open the connection.udl file you just edited. Copy the complete 3rd line of this file which contains the database connection information.
- 6) Click on *START > PROGRAMS > ASP Store Locator > ASP Store Locator*. NOTE : Steps 6 thru 8 or not necessary to get the product functional, but are highly recommended to test and debug easily.
- 7) Paste the connection information that is currently in memory into the *Connection String* text box found at the top of the application.
- 8) Test the connection by clicking on *Execute* button on any of the example queries. You should see test records results fill the results at the top of the page. If records are returned then the DLL and Database were installed correctly. If no records are returned, click on the *Get Error* Message for clues into the problem
- 9) Next navigate to the Locator folder and open the file name "inc_constants.asp" with Notepad. Set the strCon string equal to the connection string by pasting the value between double quotes of the strCon line.

EXAMPLE :

```
strCon = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\databases\storelocator.mdb;Persist Security Info=False"
```

- 10) Directly below that are two more variables : "strStoreTableName", "strZIPSTableName". The variable "strStoreTableName" is the name of the table which this instance of the locator will be looking up to find the store/retailers information. Its default settings is set to "Retailers", do not edit for this installation, but make a mental note for future installations. The second variable is "strZIPSTableName", which is the name of the ZIP code table which contains all the ZIP Codes of the United States. You should not have to edit this unless you change the name of the Zips table in the database.
- 11) Save and close the file.
- 12) Move the entire Locator folder into your web site root folder.

- 13) Test the sample pages by opening your web site in your browser and navigating to the /locator folder.
Example : <http://www.yoursitename.com/locator>

NOTE :In the future if you upgrade the DB to SQL Server use Microsoft OLEDB Provider for SQL Server as the provider

SERVER VERSION ONLY

ADDING ANOTHER INSTANCE OF THE STORE LOCATOR TO A WEBSITE

- 1) Copy the Data Structure of the Retailers table and name the table appropriately.
- 2) Copy the locator folder from the first installation into the wwwroot of the new site.
- 3) Open the "inc_constants.asp" file for modification.
- 4) Give the AdminPassword a new value
- 5) Set the "strStoreTableName" variable to the name of the table you added in step 1.
- 6) Test the sample pages by opening your web site in your browser and navigating to the /locator folder.
Example : <http://www.yoursitename.com/locator>

COM Component

The following functions can be called after instantiating the ZipCodeLocator Locator Object.

Example : (In VisualBasic) Set zipLoc = New ZipcodeLocator.Locator
(In VBScript) Set zipLoc = Server.CreateObject("ZipcodeLocator.Locator")

Properties

ConnectionString (*DataConnectionString as string*)

Description: Sets the database connection information for the COM Object. **This value must be set before calling any methods, otherwise they will fail.**

<i>Parameter</i>	<i>Description</i>
DataConnectionString	The connection information to the database.

EXAMPLE :

zipLoc.ConnectionString("Provider=Microsoft.Jet.OLEDB.4.0; Data Source=C:\Projects\zipcoded\ldb\storelocator.mdb;Persist Security Info=False")

SERVER VERSION ONLY

StoreTableName (*RetailTableName as string*)

Description: Sets the name of the table in the database from which the store information will be pulled from. **This value must be set before calling any methods, otherwise they will fail.**

<i>Parameter</i>	<i>Description</i>
RetailTableName	The name of the table that contains the store information.

EXAMPLE :

zipLoc.StoreTableName("Retailers")

SERVER VERSION ONLY

ZIPCodesTableName (*ZIPSTableName as string*)

Description: [Used Rarely] Sets the name of ZIP Codes table. Use only when you edit the name of the default Zips Table from the default name "Zips"

<i>Parameter</i>	<i>Description</i>
ZIPCodesTableName	The name of the table that contains the zip codes

EXAMPLE :

zipLoc.ZIPCodesTableName ("PostalCodes")

ErrorString()

Description: Contains useful information about the error if one occurs while running the COM Object. You should check to make sure this is empty after using any of the methods.

EXAMPLE :

ErrorString = zipLoc.ErrorString()

Methods

CalculateLocations(*ZIPCode as string, NumberOfReturnRecords as integer, OPTIONAL StringOfExtraFields as string, OPTIONAL UnitsOfMeasure as string, OPTIONAL FieldName as string, OPTIONAL FieldValue as string*) as recordset

Description: Returns the closest locations to a given ZIPCode. The default return set includes the following fields found in the retailers table: ID, Distance, Address, City, State, PostalCode, PhoneNumber, Extension, CompanyName, and EmailAddress.

Parameter	Description
ZIPCode	Postal Code of search location
NumberOfReturnRecords	The number of records to return
StringOfExtraFields	Comma separated field names you want to be returned that are not part of the default return set.
UnitsOfMeasure	Accepts Miles(<i>default</i>) or kilometers
FieldName	Name of field you want to filter the results by
FieldValue	Value of field you want to filter the results by

EXAMPLE :

```
zipLoc.CalculateLocations("98332",10,"Field1, Field2","miles")
```

CalculateLocationsByRadius(*Radius as double, ZIPCode as string, MaxNumberOfReturnRecords as integer, OPTIONAL StringOfExtraFields as string, OPTIONAL UnitsOfMeasure as string, OPTIONAL MinRadius as double*) as recordset

Description: Returns all the locations that fall within the given Radius. The default return set includes the following fields found in the retailers table: ID, Distance, Address, City, State, PostalCode, PhoneNumber, Extension, CompanyName, and EmailAddress.

Parameter	Description
Radius	Radius Distance you want to search for
ZIPCode	Postal Code of search location
MaxNumberOfReturnRecords	The maximum number of records to return
StringOfExtraFields	Comma separated field names you want to be returned that are not part of the default return set.
UnitsOfMeasure	Accepts Miles(<i>default</i>) or kilometers
MinRadius	Minimum radius value to return

EXAMPLE :

```
zipLoc.CalculateLocations( 50.0, "98332",100,"Field1, Field2","miles")
```

ResultsbyStringField(*FieldName as string, FieldValue as string, OPTIONAL OrderByField as string*) as recordset

Description: Returns all the records that have the FieldValue in the FieldName in the Retailers table.

Parameter	Description
FieldName	Field in Retailers Table you want to search on
FieldValue	Value you want to search for
OrderByField	Field you want to Order By (<i>default</i> : City)

EXAMPLE :

```
zipLoc.ResultsbyStringField("State","WA","City") or  
zipLoc.ResultsbyStringField("Cuisne","Mexican","City")
```

DistanceBetweenTwoLonLat(*Long1 as double, Lat1 as double, Long2 as double, Lat2 as double*) as double

Description: Returns the distance between two points given their Longitude and Latitude.

Parameter	Description
Long1	Longitude of point 1
Lat1	Latitude of point 1
Long2	Longitude of point 2
Lat2	Latitude of point 2

EXAMPLE :

zipLoc.DistanceBetweenTwoLonLat(118.4109,34.0889,122.3345,47.6048)

DistanceBetweenTwoZips(*ZipCode1 as string, ZIPCode2 as string*) as double

Description: Returns the distance between two points based on their ZIP Code.

Parameter	Description
ZipCode1	Postal Code of point 1
ZIPCode2	Postal Code of point 2

EXAMPLE :

zipLoc.DistanceBetweenTwoZips("90210","98174")

ResultDetails (*ID as integer*) as recordset.

Description: Returns the record that has this ID value in the Retailers table.

Parameter	Description
ID	ID of a record in Retailers table

EXAMPLE :

zipLoc.ResultDetails(17)

SQLPassThrough (*SQLStatement as string*) as recordset.

Description: Allows you to pass in any SQL statement you like. Use it to do all other queries not possible with the other methods.

Parameter	Description
SQLStatement	A valid SQL statement

EXAMPLE :

zipLoc.SQLPassThrough("SELECT TOP 10 FROM Retailers WHERE City='Seattle' and State='WA' ") or
zipLoc.SQLPassThrough("SELECT TOP 10 FROM Retailers WHERE City='& strCity &' and State='& strState &' ")

TROUBLE SHOOTING

QUESTION : I received the following error message on the sample pages :

Server object error 'ASP 0178 : 80070005'
Server.CreateObject Access Error
/locator/resultsbyzip.asp, line XX
The call to Server.CreateObject failed while checking permissions. Access is denied to this object.

ANSWER : It is a known bug from Microsoft. Read the Microsoft Knowledge Base Article - 278013 for the fix.

<http://support.microsoft.com/default.aspx?scid=KB;en-us;q278013>

QUESTION : I received the following error message on the admin pages :

Microsoft JET Database Engine error '80004005'
Operation must use an updateable query.
/locator/admin/retail_process.asp, line XX

ANSWER : The IUSR_ *computername* account does not have sufficient privileges to make edits to your Access database. Assign write privileges to this file for IUSR_ *computername*.

Upgrading to SQL server

1. Create a new Database in SQL
2. Use the Import/Export Wizard in SQL to pull in the two tables from Access.
3. Verify the Retailers table has ID as Primary Key and is set to auto increment in the SQL DB.
4. Create a UDL file by creating a new text file then rename the extension to UDL.
5. Double click on the file then click on the Provider tab and choose "Microsoft OLEDB Provider for SQL Server"
6. Click on the Connection tab and fill in the necessary information.
7. Click Test Connection.
8. Upon success apply changes then close down the file and open it up in Notepad.
9. Copy the third line into memory and close the file.
10. Open the "inc_constants.asp" page in Notepad found in the "locator" folder.
11. Paste the connection information in memory between the two quotes.

Example:

```
strConn = "Provider=SQLOLEDB.1;Password=ZipCodeUser;Persist Security Info=True;User ID=ZipCodeUser;Data Source=localhost"
```

12. Save the file and test

FAQ

Q: You provide a Access Database, but can I use another Database instead?

A: Yes, just import the data into your database and change the strConn line of code in the dbconnection.asp file. It just needs to be an ODBCcomplaint database.

MS SQL SERVER EXAMPLE:

```
strConn = "Provider=SQLOLEDB.1;Password=ZipCodeUser;Persist Security Info=True;User ID=ZipCodeUser;Data Source=localhost"
```

Q: Will it work on my Windows workstation?

A: No, it will only run on Windows NT 4.0 server or above with IIS 4.0 or above installed.

Q: Can I register the your .DLL using COM+?

A: Yes. We suggest you create a new Windows account to be used for the authentication in the COM+. That way the COM object permissions level is easy to maintain and debug security issues. If you choose to use the default account the component will be run under the System User account.

Q: I have added all my stores in the backend (not using the provided admin tool), but need to update the Lon and Lat values. How do I do it?

A: Simple just run the following query in Access :

```
UPDATE Retailers
SET Retailers.Lon = [Zips].[lon], Retailers.Lat =[Zips].[lat]
FROM Retailers, Zips
WHERE (((Retailers.PostalCode)=[Zips].[Zip]));
```

After the query is done validate these columns in the Retailers table to make sure they have data. If values are missing most likely the zip code is wrong for that store. A small percentage may not be covered in the Zips 43,000 records and in those cases just update with a nearby cities Lon and Lat.

If you are using SQL server then you can use this Stored Procudure to do all the work:

```
CREATE PROCEDURE spUpdateLonLat
    @StoreTableName nvarchar(500) = 'Retailers'
AS

DECLARE @SQL AS nvarchar(750)
Set @SQL= 'UPDATE '+@StoreTableName+' SET '+@StoreTableName+'.Lon =
Zips.[lon], '+@StoreTableName+'.Lat =Zips.[lat] FROM '+@StoreTableName+',
Zips WHERE '+@StoreTableName+'.PostalCode=Zips.[Zip]'
EXEC (@SQL)
GO
```