Product Technical Overview

VIASAT GeoTechnologies

EZTag Mobile Software Development Kit

February 2005

An Open Environment for the Development of Mobile GPS/GIS Applications

The **EZTag Mobile Software Development Kit** enables software developers to add mobile GPS/GIS functionality to their existing applications and to develop new applications for GPS/GIS mobility.

The **rich feature set of EZTag CE** allows developers to use proven GPS/GIS functionality and focus on fine-tuning their mission-critical business logic. The **open architecture** of the EZTag Mobile SDK allows developers to write mobile GPS/GIS applications using the language of their choice on the **Microsoft .NET** (C#, VB.NET) or **COM** (VB, eVB, C++, eVC++, HTML/JScript) programming models. The open architecture also means that developers are free to develop on the database of their choice, including but not limited to **Oracle Lite**, **Microsoft SQL Server CE**, **ESRI Shapefile** and the VIASAT GPS/TAG relational database.

This paper describes the key features and benefits of the EZTag Mobile SDK and the options available to developers for integrating existing applications and for developing new applications that take advantage of the GPS/GIS features of EZTag CE.

Key Features

- **GPS data collection**, ready for GPS post-processing
- **GPS navigation and satellite views**
- **Map Viewing** with GPS tracking
- Open choice of database including **Oracle Lite** and **SQL Server CE**
- Option to use built-in support for **ESRI Shapefile** and VIASAT GPS/TAG formats
- **Open** development environment
- Full support for the latest **Microsoft .NET** technology on desktop and devices

Key Benefits

- **Lower development cost.** The cost of adding GPS/GIS capabilities to existing applications that have no support for GPS/GIS is low. Developers can focus on business logic and rely on a proven platform for high-end GPS capabilities.

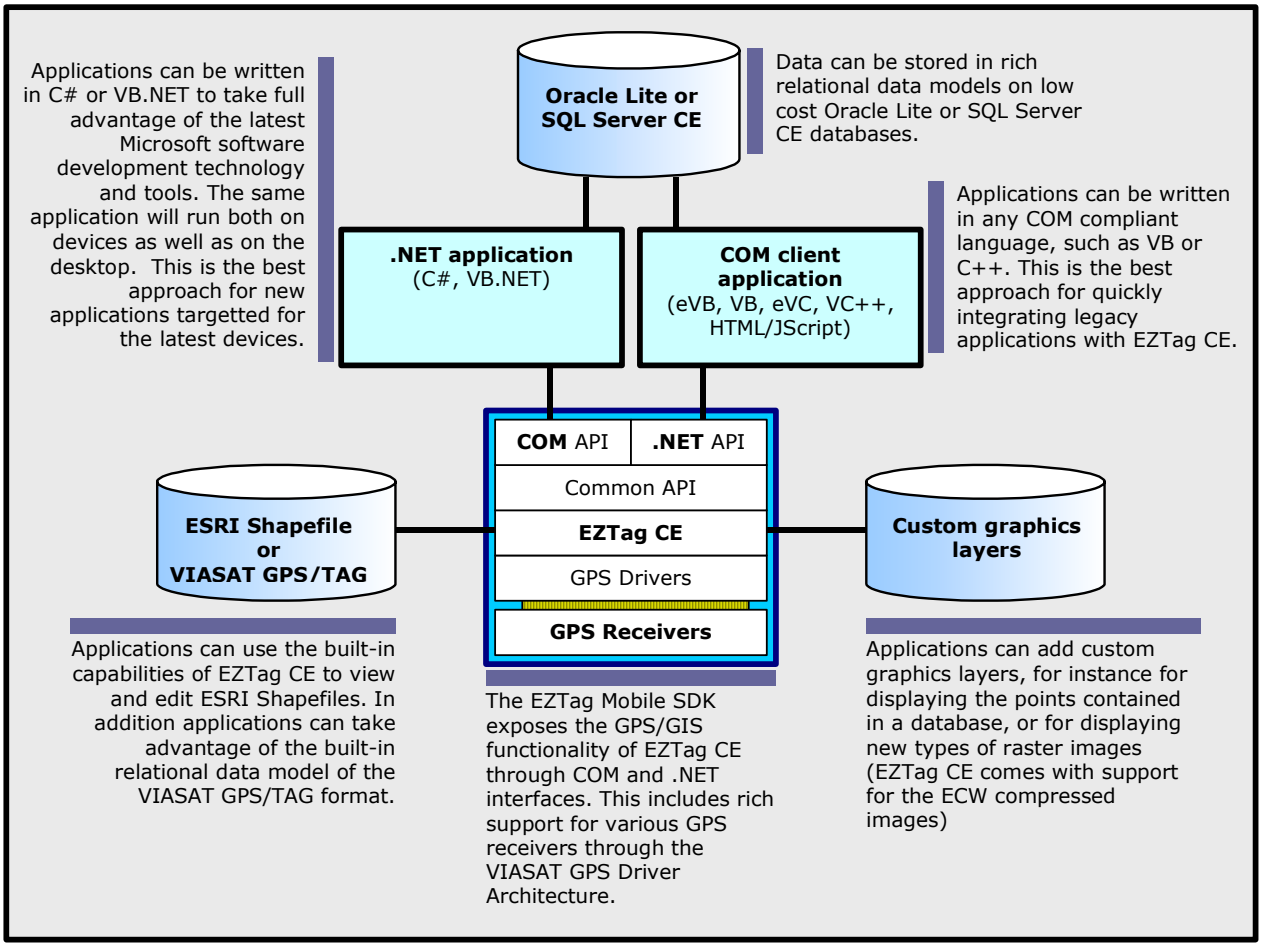
- **Longer application lifetime.** Support for the .NET programming model means that applications written today are up to date with the latest technology available on Windows and Windows CE.

- **Flexible migration path to .NET.** Support for the COM programming model means that you can use the language of your choice if you are not yet ready for .NET, or if your applications are already written in a COM compliant language such as eVB or eVC++. You can mix .NET and COM in your solution so you can gradually move towards the latest technology.

- **Easy integration with your business.** The open architecture means that you can choose which language and database components you use in your solutions.

Overview of the Architecture

The following figure illustrates the main components of the EZTag Mobile SDK architecture. Key components of the architecture are discussed below.



Key Components of the Architecture

- **EZTag CE.** At the heart of the architecture, the EZTag CE GPS/GIS application comes ready to use out of the box. This means that you can start collecting GPS points, lines and polygon data in ESRI Shapefiles and VIASAT GPS/TAG databases without any programming.
- **GPS Drivers and GPS Receivers.** The GPS Drivers enable EZTag CE to run on top of a wide variety of GPS receiver brands and models. The VIASAT GPS Driver Architecture allows you to quickly add a new type of receiver when necessary.

- **COM and .NET APIs.** The COM and .NET APIs enable application programmers to customize the EZTag CE user interface, to add custom control bars, and to add forms to respond to property queries on selected objects from ESRI Shapefiles, VIASAT GPS/TAG databases and others. User interfaces for your applications can be programmed in any COM compliant or .NET language supported by your target platform. User interfaces can also be written as Internet Explorer Web applications in HMTL/Jscript.

- **ESRI Shapefiles.** EZTag CE natively displays and edits ESRI Shapefiles. The APIs include methods to control the display of ESRI Shapefiles as ESRI ArcXML maps. The APIs also include methods to read and write attribute data associated with selected Shapefile elements.

- **VIASAT GPS/TAG databases.** EZTag CE natively displays and edits VIASAT GPS/TAG databases. The APIs include methods to read and write attribute data associated with selected GPS/TAG database elements. EZTag CE includes a dictionary editor for modeling the VIASAG GPS/TAG database without programming, which is a great approach for quick prototyping of applications, or for incremental development.

- **Relational databases (Oracle, SQL Server, etc..).** While data may be stored in ESRI Shapefile and VIASAT GPS/TAG databases, the APIs provide the methods required by applications to maintain a link between their graphic data stored in ESRI Shapefiles or VIASAG GPS/TAG databases, and their descriptive data maintained in Oracle Lite or SQL Server CE, or any other database formats. Access to the database is enabled directly through your application code.

- **Custom graphics layers.** When ESRI Shapefiles and VIASAT GPS/TAG databases are not appropriate to hold your graphics data, you may elect to develop your own graphics layer extension; for instance, to display data extracted from the relational database, or for displaying data from graphics formats that are not supported natively by EZTag CE.

Overview of the EZTag CE API

The COM and .NET APIs include the following categories:

- Forms
- Toolbars
- UI customization
- Events
- Graphics display
- GPS
- Cartographic projections

Forms

A form can be used to view or edit the attributes of an object. A form is usually opened when a user places a new object graphically with the Place tool or using the GPS, or when the user clicks the Properties menu to view the properties of an object.

The API supports two types of forms: HTML forms and Application forms. Both are similar in functionality but differ in the programming language used to code the form's logic.

In an HTML form, the logic is usually written in JScript language. An HTML form may also be designed to have logic on a Web server, in which case that logic may be written with any Web server development tool such as ASP or ASP.NET.

In an Application form, the logic is written in a COM compliant language such as VB or C++, or in a .NET language such as VB.NET or C#.

The API exposes methods that allow a form to inspect and modify the attributes of the object being viewed or edited. For instance, the object might be a line in an ESRI Shapefile, and your form might be interested in storing or retrieving a DBLINK attribute to establish a link with an Oracle Lite relational database.

Toolbars

An application can add one or more application-defined toolbars to the EZTag CE user interface . Each toolbar can be docked either on the top, left, bottom or right of the EZTag CE mainframe window. The API supports two types of toolbars: HTML toolbars and Application toolbars. Both are similar in functionality but differ in the programming language used to code the logic of the toolbar.

A sample use of a toolbar is to display the street name of the street closest to the position of the observer, based on GPS input. The toolbar logic can check periodically for the current GPS position using the API and use an application-specific search to find the closest street in a database then display the result.

UI Customization

An application can choose which user interface elements of EZTag CE will be visible and which ones will be hidden. Used together with toolbars, this makes it possible to tailor the user interface to the specific needs of the application. The EZTag CE user interface can be customized at startup using a configurable startup script written in JScript. The startup script can use the APIs to configure forms, toolbars and other UI elements.

GPS

The APIs include methods to control GPS navigation to a point. For instance, a custom toolbar might provide a list of potential navigation destinations and start navigation to a destination selected by the user from the list. The navigation compass available in the EZTag CE Plan View and Navigation View would then lead the user to the destination.

The APIs also include methods to read the current GPS position and to subscribe to GPS messages that provide up-to-date GPS information periodically, such as position information, satellite observations, etc. This can be used to build advanced GPS applications that need to use all the data coming from the GPS receiver.
Applications can also send messages to the GPS receiver; for instance, in order to change the receiver's mode of operation.

Events

The APIs include a number of events that the application can handle to provide custom functionality for those events. For instance, when the user selects an object, an event is raised to inform the application. The application can in turn use the API to read some attributes from the object and to display those attributes in a custom toolbar.

Graphics Display

The viewing extent of the graphics display can be automated by a toolbar or a form. For instance, a toolbar might include a list of cities and fit the map of the city on screen when it is selected from the list. The APIs also include a mechanism to load ESRI Shapefiles as ArcXML maps.

Cartographic Projection

The APIs contain methods to convert back and forth between geographic (lat, lon) and projected (x,y) coordinate systems. The parameters of the geographic and projected coordinate systems can be defined with EZTag CE.

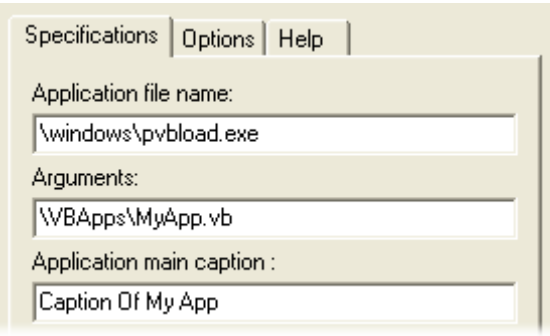
Sample Application Integration Strategies

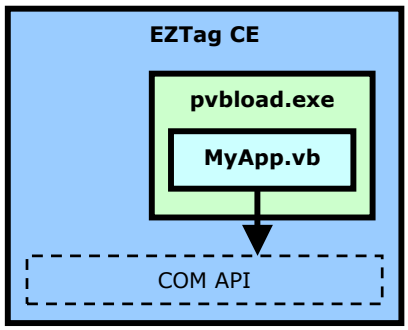
Integrating an Existing eVB Application

Context : You have a data collection form application written in eVB that has been running faithfully on a Windows CE 3.0 device since it was deployed it in the field. The application handles all the details of viewing, editing and capturing mission-critical descriptive data. While the application runs on a Windows CE 3.0 device, a separate dedicated GPS device is used for coordinates readouts and GPS navigation. The operator typically reads the coordinates from the GPS device and types them manually in the data collection form.

Problem : How do you integrate GPS functionality into the application so that the operator can carry only the Windows CE 3.0 device in the field, and so that the application automates GPS coordinates entry and integrates GPS navigation functionality?

Integration Strategy: The best strategy here is to stick with the existing application as is and modify it to use the EZTag Mobile SDK in order to accept GPS coordinates from a receiver that you will connect directly to the Windows CE 3.0 device. In general, this is fairly easy to do with the EZTag Mobile SDK as only a few method calls are required in order to interface with the GPS. The GPS navigation features are already built into EZTag CE, so you don't need to write code for that. Since EZTag CE can host external applications such as eVB applications, there is typically very little code to write in order to make your application a part of EZTag CE.





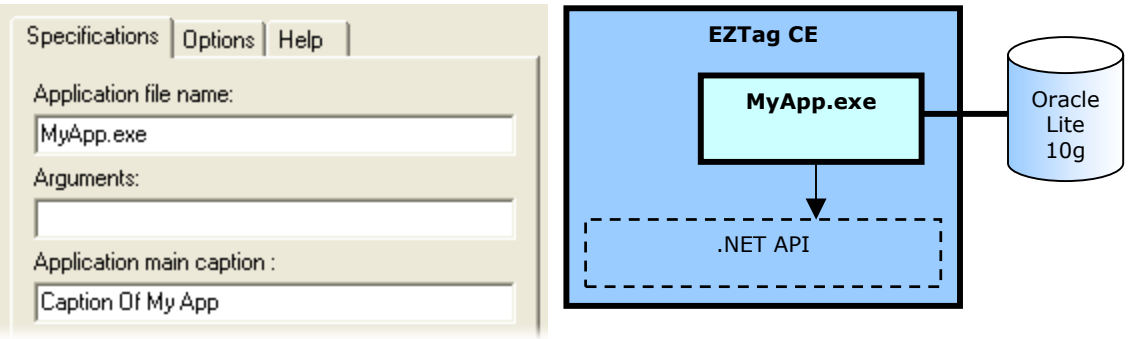
The EZTag CE dictionary editor allows you to specify that your eVB application must be loaded using the eVB loader called **pvbload.exe**. When an object is being viewed or edited, EZTag CE will automatically load your form, called **MyApp.vb** in this example. Your form will then have access to the COM API of the EZTag CE Mobile SDK in order to interact with EZTag CE and get access to the GPS functionality.

Writing a .NET Application with Oracle Lite 10g

Context: You want to write a new data form collection application that will view and edit a subset of your central Oracle database, and you want your application to collect coordinates with the GPS.

Problem: How do you develop the application without having to take risks in developing GPS functionality?

Integration Strategy: Write a new form application in VB.NET or C# and use Oracle Lite 10g as your mobile database. Integrate your application into EZTag CE as an application form and use the EZTag Mobile SDK to read the GPS coordinates into your forms. Note that a similar approach can be used with SQL Server CE instead of Oracle Lite.



The EZTag CE dictionary editor allows you to specify that your .NET application is called **MyApp.exe**. Your form will then have access to the .NET API of the EZTag CE Mobile SDK, in order to interact with EZTag CE and get access to the GPS functionality.

Other Scenarios

The EZTag Mobile SDK can be used in a wide variety of other integration scenarios. Contact VIASAT GeoTechnologies to find out more about how EZTag CE and the EZTag Mobile SDK can help you resolve your mobile GPS/GIS application development issues.