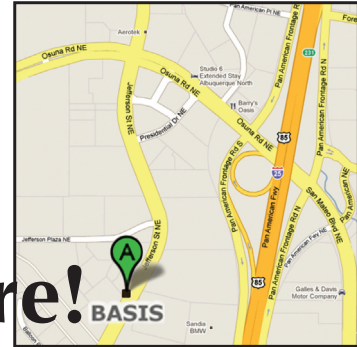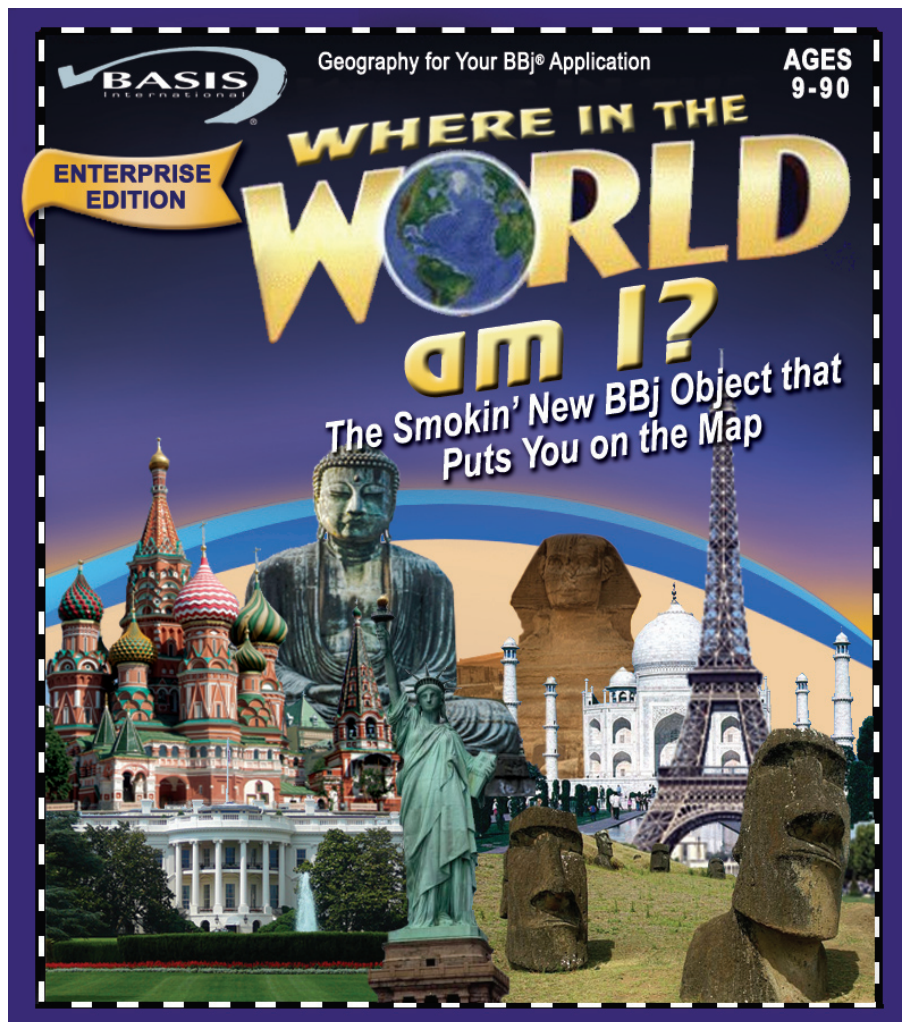# BBjGeolocation is Here!

The Global Positioning System, or GPS, technology has been booming lately. Its accuracy is even getting so precise that it is possible to pinpoint an exact location on this big planet of ours to within a few meters. This positioning capability is now built into many smart phones, wrist watches, cameras, automobiles, laptops, and, of course, dedicated GPS devices. The benefits and uses are varied and limited only by one's imagination. A few of the more popular applications of the GPS technology deal with mapping, tracking movement, and providing directions. BASIS engineers get just as excited by these new technologies as everyone else, but when the feature provides practical use cases, they start thinking about ways to integrate those features into the product line.

Since BBj is built on Java, many new capabilities are already within reach by accessing pre-built libraries, as has been the case for some of the recent additions such as e-mail and fax, PDF support, charts and graphs, Web Services, access to Google Docs, to name a few. The capability to provide location information is already built into several browsers so wouldn't that be another great feature to add to BASIS' Browser User Interface (BUI)?

Geography for Your BBj® Application
AGES 9-90
ENTERPRISE EDITION
WHERE IN THE WORLD am I?
The Smokin' New BBj Object that Puts You on the Map

GPS or Geolocation?

While both GPS and Geolocation deal with positioning, Geolocation is the automatic detection of the geographic location of a device. Geolocation focuses on providing a meaningful location and attempts to supply more specifics, when possible, by utilizing various sources for the information. GPS is just one of many lower-level back-end technologies that Geolocation utilizes to provide location information. Geolocation returns the position as a 'latitude,longitude' pair along with an estimated accuracy value. For example, the location of the BASIS offices in Albuquerque reported 35.150036,-106.593957 with an accuracy of 30 meters. **>>**

By Jim Douglas
Software Engineer

Nick Decker
Engineering Supervisor

How it Works

Desktop and mobile browsers use a combination of methods to detect the client location. The most common methods are

- GPS - requires line of sight to the GPS satellites; accurate to about 15-25 meters; sometimes slow to establish the position
- Triangulation based on cell towers (accurate to ~1000 meters)
- Triangulation based on Wi-Fi access points (accurate to about 30 meters)
- A lookup of the registered geographic location of your IP address (extremely inaccurate, used only as a last-ditch fallback)

Desktop browsers don't usually have GPS; they typically use Wi-Fi Positioning System services from Google, Skyhook Wireless, and other providers.

Geolocation API

Most current browsers support a standard Geolocation API, which returns geolocation information in a common format, regardless of how the position is determined under the covers. The BBj 11.0 BBjGeolocation API (available as a preview feature beginning in BBj 10.02) uses this standard Geolocation API and provides an interface that closely mirrors the underlying browser API.

BBjGeolocation Sample

Because it can take a while to determine the geolocation position, requests are always handled asynchronously. That is, the program requests a geolocation position, and at some point in the future, an event is fired to return the requested information or an error code if the position could not be determined. The steps are as follows:

1. Get the geolocation object (this can throw an error if geolocation is unavailable):

```
geolocation! = bbjapi().getSysGui().getGeolocation(err=unavailable)
```

2. Define an event handler for the geolocation position event:

```
geolocation!.setCallback(geolocation!.ON_GEOLOCATION_POSITION,"position")
```

3. Request the current position:

```
geolocation!.getCurrentPosition()
```

The user must approve or deny this request, since determining a user's location without their express consent may potentially compromise their privacy. Therefore, programs are never allowed to retrieve the user's geographic position without the user's explicit permission.

The geolocation position (or an error code) is returned to this event handler:

```
position:
  position! = bbjapi().getSysGui().getLastEvent()
  if position!.getStatus() = 0 then
    latitude = position!.getLatitude()
    longitude = position!.getLongitude()
    accuracy = position!.getAccuracy(); rem ' metres
    rem ' do something with the position
  endif
return
```

The complete sample program is shown in **Figure 1**. **>>**

```
rem 'Geolocation Position
   precision 6
   sysgui! = bbjapi().openSysGui("X0")
   title$ = "Geolocation Position"
   window! = sysgui!.addWindow(10,10,350,100,title$,$00090003$)
   window!.setCallback(window!.ON_CLOSE,"eoj")
   label$ = "Position (latitude,longitude):"
   label! = window!.addStaticText(100,25,25,320,25,label$,$$)
   status$ = "Waiting for position..."
   text! = window!.addStaticText(101,25,50,320,25,status$,$$)
   font! = sysgui!.makeFont("Courier New",12,sysgui!.PLAIN)
   label!.setFont(font!)
   text!.setFont(font!)
   gosub init

rem 'Process Events
   process_events

eoj:
   release

init:
   geo! = sysgui!.getGeolocation(err=unavailable)
   geo!.setTimeout(30)
   geo!.setCallback(geo!.ON_GEOLOCATION_POSITION,"position")
   geo!.getCurrentPosition()
return

unavailable:
   text!.setText(errmes(-1))
return

position:
   position! = sysgui!.getLastEvent()
   status = position!.getStatus()
   switch status
      case 0; rem 'success
         latitude = position!.getLatitude()
         longitude = position!.getLongitude()
         position$ = str(latitude)+","+str(longitude)
         text!.setText(position$)
         break
      case 1; rem 'user refused permission
         text!.setText("Failed (permission denied)")
         break
      case 2; rem 'couldn't calculate position
         text!.setText("Failed (position unavailable)")
         break
      case 3; rem '30-second timeout reached
         text!.setText("Failed (timeout)")
         break
      case default; rem 'unknown error
         text!.setText("Failed (status "+str(status)+")")
         break
   swend
return
```
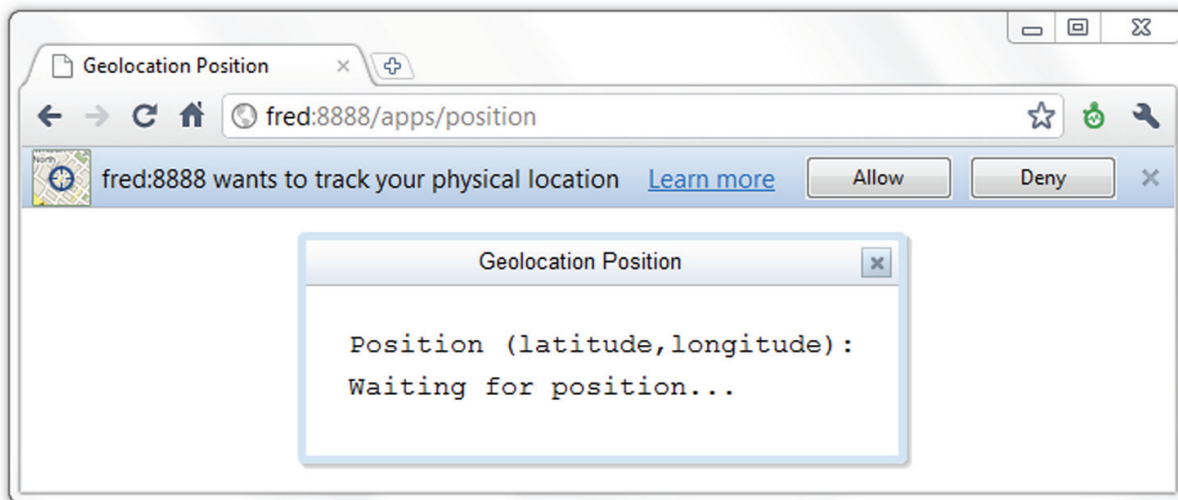
Figure 1. BBjGeolocation sample

>>

When a program wants to track the user's position, the browser must get permission from the user. **Figures 2a** and **2b** show how this looks in Chrome on Windows 7 and in Mobile Safari on iPhone, respectively.

Figure 2a. Geolocation requesting permission (Chrome, Windows 7)

Figure 2b. Geolocation requesting permission (Mobile Safari, iPhone)

If the user denies the request or if an error or timeout occurs, an error status is returned in a BBjGeolocationEvent, as shown in **Figure 3**.

If the user allows the request, geolocation position information is returned in a BBjGeolocationEvent, as shown in **Figure 4**.

To verify the accuracy of the reported position, you can hand it off to Google Maps. For example, this shows the location of the BASIS office in Albuquerque:

http://maps.google.com/?q=35.150036,-106.593957

This sample requests a single geolocation position update with the following two lines of code:

```
geolocation!.setCallback(geolocation!.ON_GEOLOCATION_POSITION,"position")
geolocation!.getCurrentPosition()
```

To receive regular position updates, substitute this single line of code in place of those two lines of code:

```
geolocation!.setCallback(geolocation!.ON_GEOLOCATION_WATCH,"position")
```
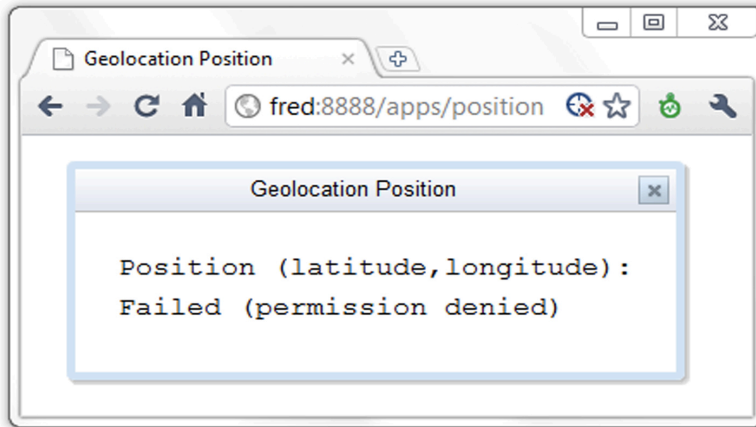
> >

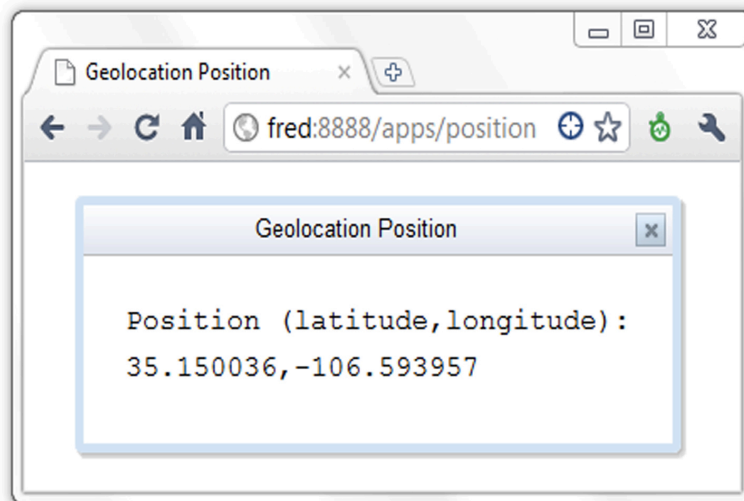**Figure 3.** Geolocation reporting that the user refused permission

Figure 4. Geolocation reporting position

Summary

The proliferation of smart mobile devices with fully capable web browsers means that now your Business BASIC application can run almost anywhere at any time. By leveraging BASIS' BUI technology, your applications can run on dozens of mobile devices such as Apple's iPad and iPhone, Google Android devices, and dozens of others that come equipped with full-featured browsers. As BUI capabilities continue to expand, so do the potential features and abilities of your application. With the latest inclusion of Geolocation, you can now write a BBx program that manages real-time tracking of fleet vehicles, integrates with Google Maps, and provides other location-based services.

Give this latest feature a try and see how easy it is to answer at least the first part of the age-old question of "Where am I and why am I here?" ■
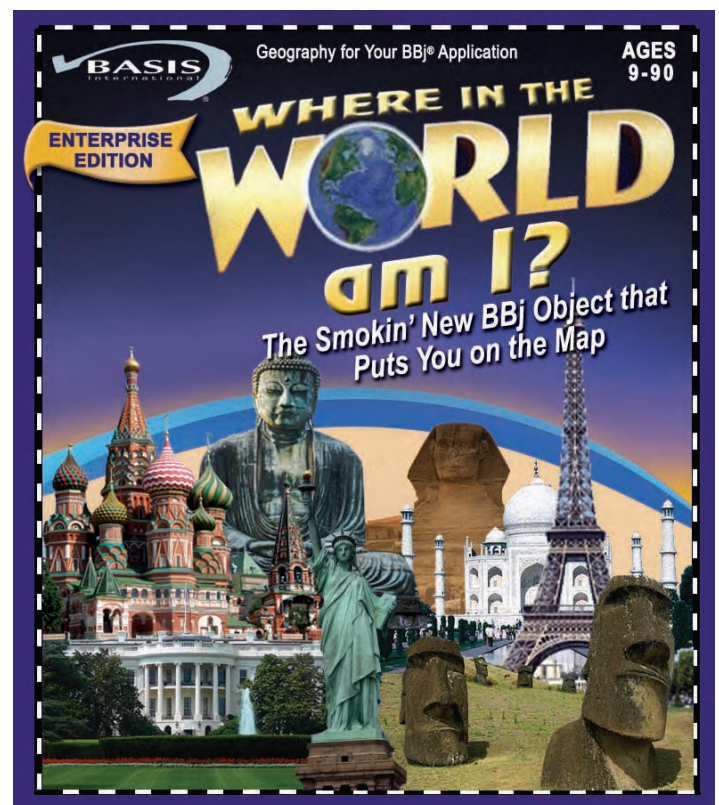
For more information about Geolocation, read these information links:
- BBjGeolocation in the online BASIS documentation
- Location-Aware Browsing test page and FAQ from Mozilla
- How to use the W3C Geolocation API from Dev.Opera
- Geolocation from Dive in to HTML
- Geolocation API Specification from W3C

Try the sample at links.basis.com/geocode