Deliver a Package Tracking System

By Brian Hipple

With end users in the software, hardware, or services industry, the ability to track the shipment of packages is an essential business function. E-commerce opportunities in the international marketplace are ever increasing, while the importance of shipping and logistical arrangements is steadily growing, especially to smaller businesses. A sale can hinge on the fulfillment of these very delivery arrangements. One simple, yet effective way to make sure a product ships on time and to the right location is to add package tracking to your application.

Web Services Basics

Whether you develop a BBj® Character User Interface (CUI) or a Graphical User Interface (GUI) application, it can easily consume a Web service to track shipments. Several articles in previous editions of the *BASIS International Advantage* provide the detail for the general steps of how to consume a Web service.

1. Determine which Web service your application will consume and obtain the URL for the Web Services Description Language (WSDL).

2. Build the client side stubs by pointing the Web service tool of choice to the WSDL (JWSDP/AXIS/AXIS2, etc.).

3. JAR the classes created in the previous step and add this JAR to the BBjServices class via the Enterprise Manager.

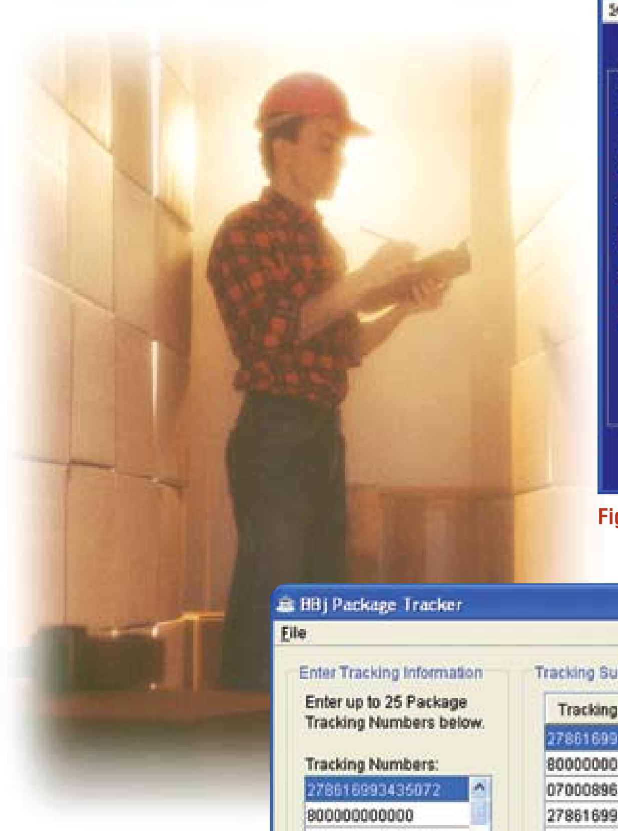
4. Add the small amount of necessary Java code to the BBj application.

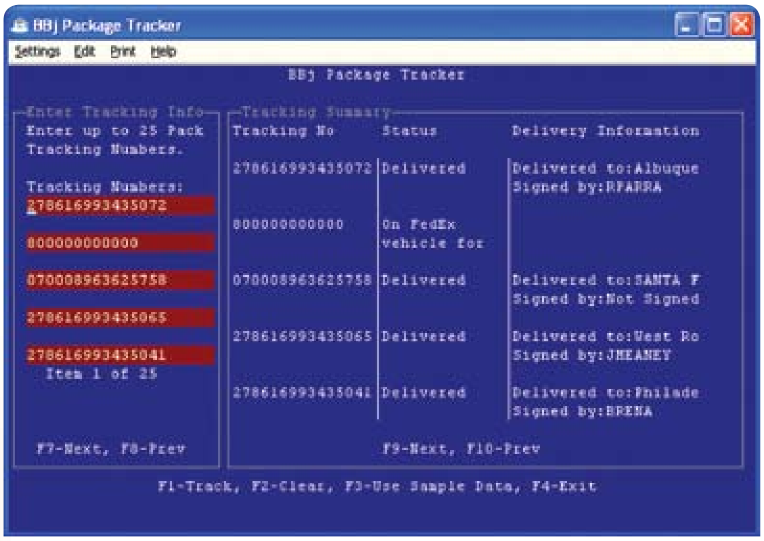
What it Looks Like

BASIS created demonstration applications that consume the FedEx package tracking Web service. Like many Web services, this service is free of charge! The "BBj Package Tracker" demo† passes tracking numbers and other criteria to the Web service, which returns tracking information such as the current location of the package and delivery information, where it was delivered, and who received/signed for the package. On the next page, **Figure 1** and **Figure 2** illustrate CUI and GUI versions of this program, both of which are available for download.

continued...

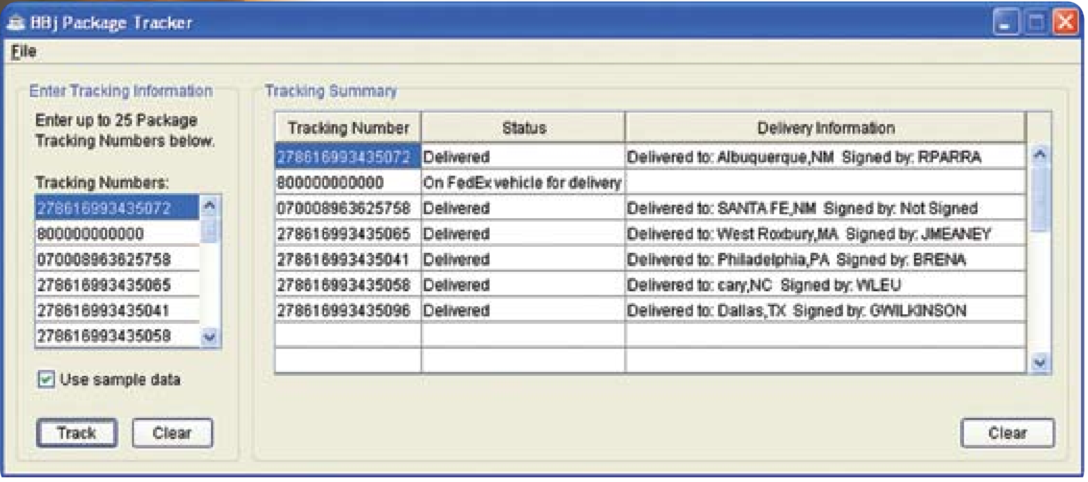
Brian Hipple
QA Test Engineer Supervisor





```
                         BBj Package Tracker

-Enter Tracking Info-  -Tracking Summary-
Enter up to 25 Pack    Tracking No      Status       Delivery Information
Tracking Numbers.
                       278616993435072 |Delivered   |Delivered to:Albuque
Tracking Numbers:                                   |Signed by:RPARRA
278616993435072
                       800000000000     On FedEx
800000000000                            vehicle for

070008963625758        070008963625758 Delivered    Delivered to:SANTA F
                                                    Signed by:Not Signed
278616993435065
                       278616993435065 Delivered    Delivered to:West Ro
278616993435041                                     Signed by:JMEANEY
  Item 1 of 25
                       278616993435041 Delivered    Delivered to:Philade
                                                    Signed by:BRENA

 F7-Next, F8-Prev              F9-Next, F10-Prev

          F1-Track, F2-Clear, F3-Use Sample Data, F4-Exit
```

Figure 1. Result of the `CUIPackTrack.src`



Enter up to 25 Package Tracking Numbers below.

Tracking Numbers:

278616993435072
800000000000
070008963625758
278616993435065
278616993435041
278616993435058

☑ Use sample data

Tracking Number	Status	Delivery Information
278616993435072	Delivered	Delivered to: Albuquerque,NM Signed by: RPARRA
800000000000	On FedEx vehicle for delivery	
070008963625758	Delivered	Delivered to: SANTA FE,NM Signed by: Not Signed
278616993435065	Delivered	Delivered to: West Roxbury,MA Signed by: JMEANEY
278616993435041	Delivered	Delivered to: Philadelphia,PA Signed by: BRENA
278616993435058	Delivered	Delivered to: cary,NC Signed by: WLEU
278616993435096	Delivered	Delivered to: Dallas,TX Signed by: GWILKINSON

Figure 2. Result of the `GUIPackTrack.src`

How it Works

The Java code necessary to consume the Web service appears in **Figure 3**, first creating an instance of the service client stub (instructed in Step 2.):

```
rem Access the tracking service
service! = new TrackService_Impl()
port!= service!.getTrackServicePort()
```

Figure 3. Create an instance of the service client stub

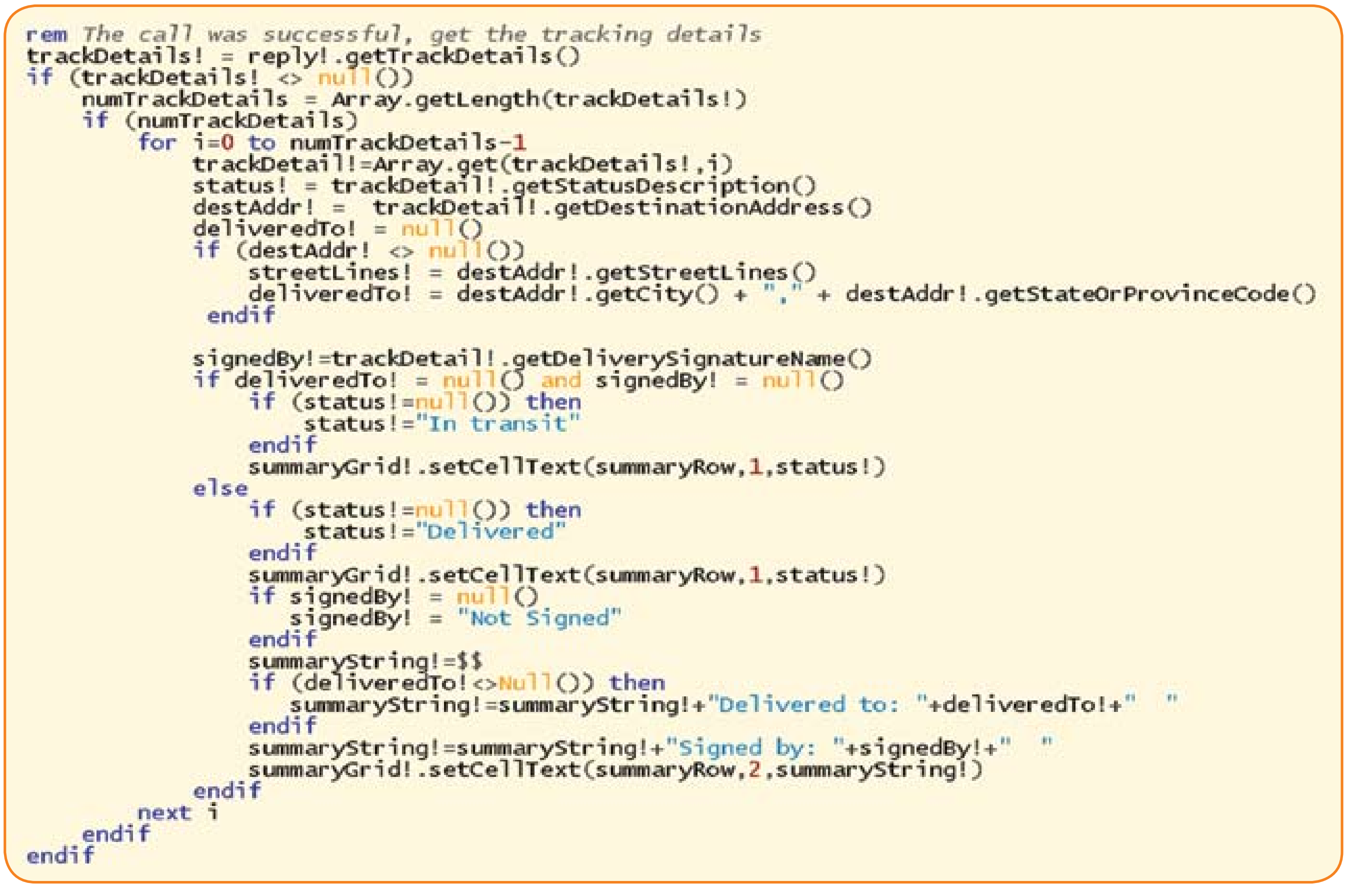
The tracking feature of the FedEx Web service is a little atypical of most Web services in that it requires a greater amount of information to invoke. The track method may look very ominous at first glance with twelve parameters, but on the positive side, it allows for broader functionality. Along with providing the tracking number, the consumer can specify criteria such as a date range, whether to include detail scans, and if the tracking information should transmit to a pager. **Figure 4** contains the BBj/Java code for tracking a shipment.

```
rem Tracking number
packageID!.setValue(package$)

reply! = port!.track(webAuthDetail!,clientDetail!,transactionDetail!,
:    versionID!,packageID!,trackingNumberUniqueIdentifier!,beginTime!,
:    endTime!,shipmentAccountNumber!,destination!,new Boolean("false"),
:    pagingToken!)
```

Figure 4. BBj/Java code for tracking a shipment

continued...

When the track method returns, the reply object reveals if it was successful. If it was unsuccessful, the program notifies the user of the failure along with error information the Web service provided. Otherwise, it extracts the tracking details from the reply and displays the tracking information to the end user, shown in **Figure 5**.



```
rem The call was successful, get the tracking details
trackDetails! = reply!.getTrackDetails()
if (trackDetails! <> null())
    numTrackDetails = Array.getLength(trackDetails!)
    if (numTrackDetails)
        for i=0 to numTrackDetails-1
            trackDetail!=Array.get(trackDetails!,i)
            status! = trackDetail!.getStatusDescription()
            destAddr! =  trackDetail!.getDestinationAddress()
            deliveredTo! = null()
            if (destAddr! <> null())
                streetLines! = destAddr!.getStreetLines()
                deliveredTo! = destAddr!.getCity() + "," + destAddr!.getStateOrProvinceCode()
             endif

            signedBy!=trackDetail!.getDeliverySignatureName()
            if deliveredTo! = null() and signedBy! = null()
                if (status!=null()) then
                    status!="In transit"
                endif
                summaryGrid!.setCellText(summaryRow,1,status!)
            else
                if (status!=null()) then
                    status!="Delivered"
                endif
                summaryGrid!.setCellText(summaryRow,1,status!)
                if signedBy! = null()
                   signedBy! = "Not Signed"
                endif
                summaryString!=$$
                if (deliveredTo!<>Null()) then
                   summaryString!=summaryString!+"Delivered to: "+deliveredTo!+"  "
                endif
                summaryString!=summaryString!+"Signed by: "+signedBy!+"  "
                summaryGrid!.setCellText(summaryRow,2,summaryString!)
            endif
        next i
    endif
endif
```

Figure 5. Extract the tracking details and display the information

Summary

Adding package tracking to your software enables end users to track shipments without having to leave your application and go to a provider's Web site. These samples should give developers a good starting point for finding out how Web services work and the functionality that they can provide. Consuming Web services in BBj is a very simple to implement yet powerful feature that any developer can take advantage of today.

Read these related *BASIS International Advantage* articles
The QA Memos Web Service
BBj and Web Services

† Download with BBj from www.basis.com/products/bbj/download.html and select "Demos" in the Optional File section. After completing the installation, select BBj > Demos > LaunchDock from the BASIS folder to run the demo.