1. Navigate to the url of the Ticket Label upload web service


You will see the following page, which contains directions on how to create the client class (in C# or VB) and the configuration files to be used to call the service.

![TicketLabelSvc Service](image1)

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following syntax:

```
```

You can also access the service description as a single file:

```
https://tat.pbid.com/FDOT_TAT.TicketLabelService/TicketLabelSvc.svc?single.wsdl
```

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to call the service. For example:

```csharp
class Test
{
    static void Main()
    {
        TicketLabelSvcClient client = new TicketLabelSvcClient();
        // Use the 'client' variable to call operations on the service.
        // Always close the client.
        client.Close();
    }
}
```

2. Call the svcutil.exe from the Visual Studio command line passing in the url from above.

![Generating files](image2)

3. The following files will be generated (Visual Studio install location may be different):
C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC\TicketLabelSvc.cs

C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC\output.config

4. Copy the configuration information from output.config into your application’s configuration file.
5. Implement in your application similar to the following example:

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using TestServices.TicketLabelServiceRef;
using TestServices.TicketServiceRef;

namespace TestServices {
    class Program {
        static void Main(string[] args) {
            // Create a new service client
            TicketLabelSvcClient client = new TicketLabelSvcClient();

            // Load the ticket values as strings
            Ticket ticket = new Ticket();
ticket.CummTons = "21.53";
ticket.GrossTons = "21.53";
ticket.LoadNumber = "1";
ticket.MixDesignGmmGsb = "Gmm";
ticket.MixDesignGmmGsbValue = "2.410";
ticket.MixDesignNumber = "SPM 09-7369A TEST";
ticket.MixDesignType = "Mix Type TEST";
ticket.NetTons = "21.53";
ticket.Plant = "A1234";
ticket.ProjectFinNumber = "123456-1-22-22";
ticket.TareTons = "21.53";
ticket.TicketIDNumber = "123458";
ticket.TicketTimeStamp = DateTime.Now.ToString();
ticket.TruckIDNumber = "0589";

            // Call the service method "Upload Single Ticket Label"
            // If there are errors they will be returned in the response
            TicketLabelServiceRef.ServiceResponse response = client.UploadSingleTicketLabel(ticket);
            client.Close();
        }
    }
}

ALTERNATIVELY:

1. Add a service reference to your project: https://tat.pbid.com/fdot.tat.ticketlabelservice
2. Use the ticket upload service as in the code example above.
The Ticket Label upload service is expecting the following fields to be populated in the ticket label (all are strings):

- TicketIDNumber
- Plant
- MixDesignNumber
- MixDesignType
- MixDesignGmmGsb
- MixDesignGmmGsbValue
- ProjectFinNumber
- TruckIDNumber
- LoadNumber
- NetTons
- CummTons
- GrossTons
- TareTons
- TicketTimeStamp