



# **T2 Flex™ 7.4 Web Services**

**API Reference**

## TABLE OF CONTENTS

|  |     |
|--|-----|
| Chapter 1: Introduction.....                             | 4   |
| About Web Services .....                                 | 4   |
| How T2 Flex Uses Web Services .....                      | 4   |
| Terminology.....   | 4   |
| Installing and Upgrading Web Services.....               | 5   |
| Configuring T2 Flex for Web Services.....                | 5   |
| Chapter 2: Using Web Services.....                       | 6   |
| Calling Web Service Methods .....                        | 6   |
| Input Parameters .....                                   | 7   |
| Return Parameters .....                                  | 9   |
| Data Types.....  | 9   |
| Using T2 Web Services to Query the T2 Flex Database..... | 10  |
| Chapter 3: Web Services Function Reference .....         | 15  |
| T2_Flex_Adjudications.....                               | 15  |
| T2_Flex_Allocations.....                                 | 22  |
| T2_Flex_BulkPermissions .....                            | 25  |
| T2_Flex_CC_PADSS.....                                    | 27  |
| T2_Flex_Contraventions.....                              | 27  |
| T2_Flex_Dockets.....                                     | 32  |
| T2_Flex_EncodingModule.....                              | 33  |
| T2_Flex_Entities.....                                    | 33  |
| T2_Flex_Events .....                                     | 40  |
| T2_Flex_HCS.....   | 49  |
| T2_Flex_IVR.....   | 52  |
| T2_Flex_LoyaltyProgram.....                              | 54  |
| T2_Flex_Maps.....  | 65  |
| T2_Flex_Misc.....  | 65  |
| T2_Flex_MiscSaleItems .....                              | 67  |
| T2_Flex_Occupancy .....                                  | 69  |
| T2_Flex_Permissions .....                                | 70  |
| T2_Flex_POSStations .....                                | 105 |
| T2_Flex_Properties.....                                  | 105 |

|  |     |
|--|-----|
| T2_Flex_Reporting.....                             | 106 |
| T2_Flex_Reservation .....                          | 108 |
| T2_Flex_ThirdParty.....                            | 111 |
| T2_Flex_ValidationProviders.....                   | 112 |
| T2_Flex_Vehicles .....                             | 119 |
| T2_Flex_Waitlists .....                            | 122 |
| T2_Flex_Wireless.....                              | 125 |
| Chapter 4: External Web Services .....             | 126 |
| External Custom Action Web Services.....           | 126 |
| External Permit Eligibility Web Services .....     | 127 |
| External Sales Web Services.....                   | 133 |
| Chapter 5: Web Services Error Messages .....       | 138 |
| Chapter 6: Miscellaneous Notes For Developers..... | 151 |
| Querying the T2 Flex Database .....                | 151 |
| Error Logging .....                                | 153 |
| Chapter 7: Known Issues List.....                  | 154 |
| T2_Flex_Entities.....                              | 154 |
| T2_Flex_Permissions .....                          | 154 |
| T2_Flex_Vehicles .....                             | 155 |

# CHAPTER 1: INTRODUCTION

This document is provided for T2 customers and business partners who wish to integrate external programs with T2 Flex™ using T2's Web Services Application Programming Interface (API). It describes the installation, configuration, and usage of the API.

This document is intended for systems administrators who are responsible for the installation and configuration of T2 Flex, and for developers who need to write programs that interact with T2 Flex data, particularly programs that need to insert or modify records in the T2 Flex database.

## ABOUT WEB SERVICES

A Web service is a specific type of program whose purpose is to allow exchange of data with other systems. The public interface to a Web service is specified in XML, and the exchange of data with a Web service also uses XML-based messages, conveyed using standard Internet protocols such as HTTP and SOAP. This use of standard protocols allows for greater interoperability between systems built on different platforms and/or using different programming languages.

Note that Web services get the “Web” portion of their name from their use of these Internet protocols; the use of “Web” in the name should not imply that Web services are only useful in developing end-user Web applications (although that is a common use).

**Note:** This document assumes the reader's familiarity with Web services, with relational databases, and with API programming in general. While this document describes T2 Flex Web services in detail, it does not attempt to describe the use of Web services in every possible programming context. Consult the documentation for your programming language of choice to determine what mechanisms exist for calling Web services. In addition, many public Internet sites that contain sample code for calling Web services in various programming languages.

## HOW T2 FLEX USES WEB SERVICES

T2 Flex Web services allow external programs to insert, update, and read records in the T2 Flex database, or to perform complicated series of database queries by issuing single commands. Each Web service offers methods, or functions, to interact with its corresponding object.

## TERMINOLOGY

For the purposes of this document the following terminology is used:

- **Web Service:** See description in **About Web Services**.
- **Web Service Method:** A method is an API function, such as **InsertUpdateEntity**, that is exposed by a Web service. [Chapter 3](#) describes, for each T2 Flex Web

service, the specific methods (functions) available to the programmer through that Web service.

- **UID:** Abbreviation for **U**nique **I**dentifier: a column found in the T2 Flex database that uniquely identifies a record.
- **Adjudication:** General term used in T2 Flex technical designs and documentation for the common synonyms **appeal** or **hearing**.
- **Contravention:** General term used in T2 Flex technical designs and documentation for common synonyms such as **citation**, **ticket**, or **violation**.
- **Entity:** General term used in T2 Flex technical designs and documentation for the common synonym **customer**.
- **Permission:** General term used in T2 Flex technical designs and documentation for the common synonym **permit**.

## INSTALLING AND UPGRADING WEB SERVICES

Download the **T2 Flex Installation and Upgrades for Self-hosted Customers** document from the T2 Systems website or, more conveniently, reference the procedures in the T2 Knowledge Base: <http://knowledgebase.t2systems.com>.

There is no separate installation process for upgrading your existing Web services installation to the current version. Follow the installation instructions above to upgrade from a previous version of Web services.

Each version of Web services corresponds to a version of T2 Flex. Do not upgrade Web services without upgrading to the corresponding version of T2 Flex, or vice-versa.

## CONFIGURING T2 FLEX FOR WEB SERVICES

T2 Flex must be configured properly in order to work with Web services. See specific instructions in T2 Flex Online Help for each of the following configuration areas:

- **User Management** – You must define least one “Web services user” so that your external program can access the T2 Flex database.
- **Cash Drawer Lookup Table Setup** - You must create at least one cash drawer for the Web services to use when processing payments for citations and/or permits.
- **Allocating Permits to a Cash Drawer** - If your external program is used to sell permits, you must first to allocate permits to the Web services cash drawer. You must only sell these permits via Web services; they should not be sold by a T2 Flex user.

**Note:** It is not necessary to segregate permits allocated for web sales by setting up separate number ranges or control groups. Unsold permits allocated to a Web services drawer can be de-allocated and re-allocated to in-office cash drawers as necessary.

## CHAPTER 2: USING WEB SERVICES

### CALLING WEB SERVICE METHODS

T2 Flex Web services use the Microsoft .NET framework, which supports three protocols for calling Web service methods: HTTP GET, HTTP POST, and SOAP.

**NOTE:** HTTP GET is disabled by default when T2 Flex Web Services are installed. For details of how to enable or disable protocols for calling Web services, see the following Web page:

<http://support.microsoft.com/default.aspx?scid=kb;en-us;819267>

The URL for each Web service can be constructed as follows:

`https://<server name>/<directory>/<servicename>.asmx`

Where **<server name>** is the server on which the Web services were installed; **<directory>** is the directory in which the Web services reside, chosen during installation; and **<service name>** is the name of the Web service.

The following T2 Flex Web services are available for customer use:

- T2\_Flex\_Adjudications
- T2\_Flex\_Allocations
- T2\_Flex\_BulkPermissions
- T2\_Flex\_Contraventions
- T2\_Flex\_Dockets
- T2\_Flex\_Entities
- T2\_Flex\_Events
- T2\_Flex\_HCS
- T2\_Flex\_IVR
- T2\_Flex\_LoyaltyPrograms
- T2\_Flex\_Misc
- T2\_Flex\_MiscSaleItems
- T2\_Flex\_Occupancy
- T2\_Flex\_Permissions
- T2\_Flex\_Properties
- T2\_Flex\_Reporting
- T2\_Flex\_Reservation
- T2\_Flex\_ThirdParty
- T2\_Flex\_ValidationProviders
- T2\_Flex\_Vehicles

- T2\_Flex\_Waitlists

The following Web services deal primarily with communication between hardware devices.

- T2\_Flex\_EncodingModule
- T2\_Flex\_CC\_PADSS
- T2\_Flex\_Maps
- T2\_Flex\_POSStations
- T2\_Flex\_Wireless

So, for example, on a server named `www.t2flex.com`, with Web services located in the default T2WS directory, the URL for the T2\_Flex\_Waitlists Web service would be:

`https://www.t2flex.com/PowerParkWS/T2_Flex_Waitlists.asmx`

You can browse to the URL for the Web service to see a Web page that lists the different methods available in that Web service. From this page you can also view the WSDL (Web Services Definition Language) for the Web service. Clicking on any of the method names listed will show sample SOAP requests and responses for that method.

Starting with T2 Flex version 7.2, web methods do not use primitive types. Rather, the information is encapsulated in a special wrapper object (complex type). Users will not be able to use the web method from the discovery page but will instead need to use a suitable test program such as *soapUI*.

## INPUT PARAMETERS

Each Web service method takes four parameters as described in the table below.

### Parameter List

| Parameter       | Description  | Required |
|-----------------|--|----------|
| <b>Xmldata</b>  | A XML formatted string containing additional parameters specific to the method being called. The expected format of the Xmldata parameter will differ from one method to another as described below. | Yes      |
| <b>Version</b>  | Web service method version.  | Yes      |
| <b>Username</b> | T2 Flex username for Web services user.  | Yes      |
| <b>Password</b> | T2 Flex password for Web services user.  | Yes      |

### The Xmldata Parameter

For each Web service method, the format of the Xmldata parameter is an XML formatted string that contains all parameters specific to the method being called. [Chapter 3](#) describes each Web service method and its required and optional parameters. For example, using the **InsertUpdateVehicleAssociation** method of the

T2\_Flex\_Vehicles Web service to perform an update, one might pass the following XML in the Xmldata parameter:

```
<?xml version="1.0" encoding="UTF-8"?>
<T2Method>
  <Parameter>
    <name>Association_UID</name>
    <value>2000</value>
  </Parameter>
  <Parameter>
    <name>Assoc_Type</name>
    <value>1</value>
  </Parameter>
  <Parameter>
    <name>Assoc_Start_Date</name>
    <value></value>
  </Parameter>
  <Parameter>
    <name>Assoc_End_Date</name>
    <value>' '</value>
  </Parameter>
  <Parameter>
    <name>Assign_Contraventions</name>
    <value>0</value>
  </Parameter>
</T2Method>
```

Among the parameters passed in the Xmldata string, if a parameter is not sent, or is sent without a value, the parameter will be ignored. If a required parameter is omitted, the Web service method will return an error.

If updating a record, a value of ' ' (single quote-space-single quote) must be sent in order to delete the existing value for a parameter. Therefore, in the sample Xmldata string above, the value of the ASSOC\_BEGIN\_DATE parameter will not be changed. However, the value of the ASSOC\_END\_DATE parameter will be cleared.

T2 Flex allows system administrators to define custom rules that govern input fields in the T2 Flex client application. For example, fields may be marked as required, or may be configured to have default values or maximum values. The Web service methods will return an error if the parameters passed for an insert or update do not conform to the constraints defined by the T2 Flex administrator. The error messages returned in such circumstances vary and are not among those listed in [Chapter 5](#).

## The Version Parameter

Each Web service method exists in one or more versions. The version parameter specifies which version of the method is being called.

The version number for a Web service method is not the same as the as the version number of T2 Flex or of the Web services installer (see [Chapter 1](#)). Each Web service method has been created at version 1.0. New version numbers are assigned only when



there has been a change to the method that affects all programs using that method. For example:

- A new or changed required input parameter in the Xmldata string
- A new or changed return parameter
- A change in the function performed by the Web service method

When a new version of a Web service method is introduced, previous versions are deprecated, meaning that they will not be maintained in the future. However, deprecated versions will continue to work for the sake of existing programs that rely on those versions. The most current version of a Web service method is always the preferred version.

## RETURN PARAMETERS

All methods will return a string containing an XML document that consists of either the output parameters resulting from a successful call, or any error messages. If the call is successful the XML document will consist of any return parameters and their values. For example, the **InsertUpdateVehicleAssociation** method would return the following string for a successful call:

```
<?XML version='1.0' encoding='UTF8'?>
<T2WsReturn>
  <Return>
    <ReturnField>Assoication_UID</ReturnField>
    <ReturnValue>2000</ReturnValue>
  </Return>
</T2WsReturn>
```

If a method call returns unsuccessfully, the XML document returned will contain any error numbers and corresponding messages. For example:

```
<T2ErrorList>
  <Error>
    <ErrorNumber>23</ErrorNumber>
    <ErrorDescription>Some error message.</ErrorDescription>
  </Error>
  <Error>
    <ErrorNumber>41</ErrorNumber>
    <ErrorDescription>Another error message.</ErrorDescription>
  </Error>
</T2ErrorList>
```

## DATA TYPES

Each of the XML parameters specified in Chapter 3 needs to be of a specific data type. The following table explains each data type used by T2 Web services.

| Data Type Label | Description                                     |
|-----------------|---|
| Integer         | A number value, typically used for UIDs.        |
| String          | A text value                                    |
| Boolean         | Either/or toggle values—Yes/No or True/False or |

|              |  |
|--------------|--|
|              | 1/0.   |
| Double       | Allows a two-digit decimal, typically used for monetary values.  |
| Complex Type | Contains one or more nested parameters, as indicated in each Web method's parameter tables.<br><b>Note:</b> Complex Types were introduced starting in T2 Flex 7.2, so most of the Web methods do not yet support them. |
| DateTime     | Indicates a date/time field.   |

## USING T2 WEB SERVICES TO QUERY THE T2 FLEX DATABASE

T2 hosted customers do not have direct Oracle access to the hosting environment for querying the database or other systems integrations tasks. In order to work around this security constraint, the ExecuteQuery() Web method allows customers to execute a T2 Flex query using Web Services.

The basic process for executing a query using Web services is as follows:

- 1) Create a query in T2 Flex with **Is Available for Web Services?** selected.

**Note:** For more information on setting up a query that will work with this Web method, see the **Query Manager Wizard** topic in T2 Flex Online Help (accessed by clicking the ? in the upper-right corner of T2 Flex).

- 2) Create an XML soap envelope that includes the query UID from T2 Flex as well as any desired parameter values. See parameter descriptions and examples below.
- 3) Run the ExecuteQuery() or ExecuteQueryByName() Web method to execute your T2 Flex query.

### ExecuteQuery() v1.0

This Web method falls under the T2\_Flex\_Misc category of Web services and allows you to execute a T2 Flex query using Web Services.

An additional ExecuteQueryByName() Web method allows you to call queries by the query name (QDE\_NAME field of the QUERY\_DEF table) instead of the query UID, reducing the risk of calling the wrong query in a production environment. The output of this Web method is the same as ExecuteQuery().

**Note:** Unlike the UID, the query name field can be altered. If this field is edited after the query is created or during the import process, the ExecuteQueryByName() Web method could fail.

#### Input XML Parameters

| Parameter       | Description  | Required           |
|-----------------|--|--------------------|
| QueryUid        | Query UID in T2 Flex   | Yes                |
| queryParameters | Contains a QueryParameter complex type* parameter set for each | Yes (one for each) |

|                |   |  |
|----------------|---|--|
|                | field/value set executed (see next item).   | parameter of the query being executed) |
| QueryParameter | Complex type* collection of the following parameters used to indicate field values that should be executed on the query: <ul style="list-style-type: none"> <li>Field (string)</li> <li>Value (string)</li> </ul> | Yes                                    |

### Return XML Parameters

| Parameter          | Description  |
|--------------------|--|
| ExecuteQueryResult | Indicates in the results string whether or not the query was executed. |

### Usage Notes

\*Complex Type – This type of parameter contains nested parameters, as indicated in the table.

## Examples of T2 Flex Queries and Corresponding XML SOAP Envelopes

In the following two examples, the UID values and parameter names are highlighted in red so you can see how the values from the query correspond with the values in the XML SOAP envelope. The parameter names, PARAM1 and PARAM2, can be whatever you want. They simply act as arbitrary placeholders in the SQL, to be replaced with the values specified in the Web service request.

Note: If you use the Query Manager wizard to create your query (as opposed to overriding the wizard with SQL), parameters are specified as **Variable** value types on the Filter step.

### Example One: Get Citations by Plate Number and Entity UID

#### Query in T2 Flex with UID 2033

```
SELECT con_uid,
       con_ticket_id,
       con_amount_due,
       ent_uid_responsible_ent,
       ent_first_name,
       ent_last_name
FROM
```

```

        contravention_view INNER JOIN entity ON ent_uid_responsible_ent
        = ent_uid)

WHERE

        CON_SNAP_VEH_PLATE_LICENSE = 'PARAM1' AND

        ENT_UID_RESPONSIBLE_ENT = 'PARAM2'

```

### XML SOAP Envelope to Execute Query

```

<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:t2s="http://www.t2systems.com/">

  <soapenv:Header/>
  <soapenv:Body>
    <t2s:ExecuteQuery>
      <t2s:version>1.0</t2s:version>
      <t2s:username>username</t2s:username>
      <t2s:password>password</t2s:password>
      <t2s:queryUID>2033</t2s:queryUID>
      <t2s:queryParameters>
        <t2s:QueryParameter>
          <t2s:Field>PARAM1</t2s:Field>
          <t2s:Value>ABC123</t2s:Value>
        </t2s:QueryParameter>
        <t2s:QueryParameter>
          <t2s:Field>PARAM2</t2s:Field>
          <t2s:Value>4500</t2s:Value>
        </t2s:QueryParameter>
      </t2s:queryParameters>
    </t2s:ExecuteQuery>
  </soapenv:Body>
</soapenv:Envelope>

```

### Sample Returned Web Service XML

```

<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ExecuteQueryResponse xmlns="http://www.t2systems.com/">
      <ExecuteQueryResult><![CDATA[<QUERY_DATASET><RECORD><CON_UI
D>2154</CON_UID><CON_TICKET_ID>TEST029</CON_TICKET_ID><ENT_
UID>0</ENT_UID><ENT_FIRST_NAME>John</ENT_FIRST_NAME><ENT_LA
ST_NAME>Smith,
Jr.</ENT_LAST_NAME></RECORD><RECORD><CON_UID>2155</CON_UID>
<CON_TICKET_ID>TEST030</CON_TICKET_ID><ENT_UID>0</ENT_UID><
ENT_FIRST_NAME>John</ENT_FIRST_NAME><ENT_LAST_NAME>Smith,
Jr.</ENT_LAST_NAME></RECORD><RECORD><CON_UID>2147</CON_UID>
<CON_TICKET_ID>TEST022</CON_TICKET_ID><ENT_UID>0</ENT_UID><
ENT_FIRST_NAME>John</ENT_FIRST_NAME><ENT_LAST_NAME>Smith,
Jr.</ENT_LAST_NAME></RECORD><RECORD><CON_UID>2156</CON_UID>
<CON_TICKET_ID>TST001</CON_TICKET_ID><ENT_UID>0</ENT_UID><E

```

```

        <ENT_FIRST_NAME>John</ENT_FIRST_NAME><ENT_LAST_NAME>Smith,
        Jr.</ENT_LAST_NAME></RECORD></QUERY_DATASET>]]></ExecuteQue
        ryResult>
    </ExecuteQueryResponse>
</soap:Body>
</soap:Envelope>

```

### Example 2: Get Customer Citations by Partial Name Lookup

#### Query in T2 Flex with UID 2034

```

SELECT con_uid,
       con_ticket_id,
       con_amount_due ent_uid,
       ent_first_name,
       ent_last_name
FROM
contravention_view INNER JOIN entity ON (ent_uid =
ent_uid_responsible_ent)
WHERE
    upper(ent_first_name) LIKE upper('PARAM1%') AND
    UPPER(ENT_LAST_NAME) LIKE UPPER('PARAM2%')

```

#### XML SOAP Envelope to Execute Query

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:t2s="http://www.t2systems.com/">
    <soapenv:Header/>
    <soapenv:Body>
        <t2s:ExecuteQuery>
            <t2s:version>1.0</t2s:version>
            <t2s:username>username</t2s:username>
            <t2s:password>password</t2s:password>
            <t2s:queryUID>2034</t2s:queryUID>
            <t2s:queryParameters>
                <t2s:QueryParameter>
                    <t2s:Field>PARAM1</t2s:Field>
                    <t2s:Value>j</t2s:Value>
                </t2s:QueryParameter>
                <t2s:QueryParameter>
                    <t2s:Field>PARAM2</t2s:Field>
                    <t2s:Value>Smith</t2s:Value>
                </t2s:QueryParameter>
            </t2s:queryParameters>
        </t2s:ExecuteQuery>
    </soapenv:Body>
</soapenv:Envelope>

```

#### Sample Returned Web Service XML

```

<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

```

```
<soap:Body>
  <ExecuteQueryResponse xmlns="http://www.t2systems.com/">
    <ExecuteQueryResult><![CDATA[<QUERY_DATASET><RECORD><CON_UID>2494</CON_UID><CON_TICKET_ID>JMH113</CON_TICKET_ID><CON_AMOUNT_DUE>0</CON_AMOUNT_DUE><ENT_UID_RESPONSIBLE_ENT>4500</ENT_UID_RESPONSIBLE_ENT><ENT_FIRST_NAME>John</ENT_FIRST_NAME><ENT_LAST_NAME>Smith</ENT_LAST_NAME></RECORD><RECORD><CON_UID>2492</CON_UID><CON_TICKET_ID>JMH111</CON_TICKET_ID><CON_AMOUNT_DUE>0</CON_AMOUNT_DUE><ENT_UID_RESPONSIBLE_ENT>4500</ENT_UID_RESPONSIBLE_ENT><ENT_FIRST_NAME>John</ENT_FIRST_NAME><ENT_LAST_NAME>Smith</ENT_LAST_NAME></RECORD><RECORD><CON_UID>2493</CON_UID><CON_TICKET_ID>JMH112</CON_TICKET_ID><CON_AMOUNT_DUE>0</CON_AMOUNT_DUE><ENT_UID_RESPONSIBLE_ENT>4500</ENT_UID_RESPONSIBLE_ENT><ENT_FIRST_NAME>John</ENT_FIRST_NAME><ENT_LAST_NAME>Smith</ENT_LAST_NAME></RECORD></QUERY_DATASET>]]></ExecuteQueryResult>
  </ExecuteQueryResponse>
</soap:Body>
</soap:Envelope>
```

## CHAPTER 3: WEB SERVICES FUNCTION REFERENCE

This section defines, for each Web service, the methods belonging to that service. For each method are listed the following:

- **Method Version:** The version number to be passed as one of the input parameters to the method (see “Input Parameters” in [Chapter 2](#)).

**Note:** The version number is listed after the method name below. For example, SellPermission() **v1.4**.

- **Input XML Parameters:** Specifies the expected format of method-specific parameters passed to the method in the Xmldata parameter (see “Input Parameters” in [Chapter 2](#)).

**Note:** Where a database field name appears in the description for a parameter, e.g., (**ENTITY.ENT\_UID**), it indicates that the value passed must match the value of this field for an existing record in the T2 Flex database.

- **Return XML Parameters:** Specifies the format of parameters contained in the XML string returned upon a successful call to the method.
- **Usage Notes:** Describes the purpose of the method and gives recommendations concerning its correct usage.

### Custom Fields

The system administrator in T2 Flex has the option of defining, for certain database objects, custom fields that are added to the Oracle data dictionary for those objects. Where the notation “(FIELD\_NAME)” appears in the parameter lists below, it indicates that the method accepts one or more custom fields as input. To specify a value for a custom field, use the name of the field as the parameter name.

**Note:** Do not enclose the field name in parentheses.

### Error Messages

The end of this chapter contains a list of error messages that may be contained in the XML string returned upon an unsuccessful call to the method. The error messages listed are organized by number and come from all Web services.

## T2\_FLEX\_ADJUDICATIONS

### ApplyAdjudicationResults() v1.0

This web method applies Appeal Results to citations or boot/tow linked to an appeal. The appeal must be open and have linked citations and/or boot/tows.

**Input XML Parameters**

| Parameter Name                    | Description   | Required?                             |
|-----------------------------------|---|---------------------------------------|
| resultsList                       | Contains an ApplyAdjudicationResultsListParam complex type* parameter set for each item on appeal (see next item).  | Yes                                   |
| ApplyAdjudicationResultsListParam | Complex type* collection of the following parameters used to apply the given results: <ul style="list-style-type: none"> <li>■ ContraventionUid (integer)</li> <li>■ BootTowUid (integer)</li> <li>■ BootTowResolveReasonUid (string)</li> <li>■ AdjudicationResultUid (integer)</li> <li>■ FineAdjustment (double)</li> <li>■ NoteCodeUid (integer)</li> </ul> | Yes (one for each item on the appeal) |
| AdjudicationUid                   | The Appeal UID to which to apply results.   | Yes                                   |
| ReinstatementDate                 | The date and time the item(s) on appeal are to be reinstated in the format of YYYY-MM-DD.   | No                                    |
| UseCalculatedReinstatementDate    | Use 1/0 values to indicate yes or no for using the calculated reinstatement date.   | No                                    |
| JudgementDecision                 | The judgment decision description.  | No                                    |
| LetterTemplateUid                 | The letter template UID to be sent with the results.  | No                                    |
| AddressUID                        | The address UID to which to send the results.   | No                                    |
| EmailAddressUID                   | The email address UID to which to send the results.   | No                                    |
| LetterOutputPDF                   | Use 1/0 values to indicate yes or no for the email format as PDF or standard HTML.  | No                                    |

**Returned XML Parameters**

| Parameter Name                 | Description   |
|--------------------------------|---|
| ApplyAdjudicationResultsResult | Indicates with a 1 or 0 in the results string whether or not the results were successfully applied to the given citation(s) and/or boot/tow(s). |



### Usage Notes

- \*Complex Type – This type of parameter contains nested parameters, as indicated in the table.
- Appeal results can include a fine reduction or fine increase. If no amount is provided the default will be used from the result lookup table. Fine amounts can only be overridden if the option “Is Adjust Amount Editable” is selected in the appeal result lookup.
- Fees associated with the Appeal Results are applied to the given citations and boot/tows instead of to the appeal.
- Overpayments are not supported via this web method.
- If the application of any appeal result generates an error, all work done up to that point will be rolled back and the error will be returned to the caller.

## EscalateAdjudication() v1.0

The Escalate web method escalates a given appeal to the next workflow stage if a subsequent stage exists.

### Input XML Parameters

| Parameter Name  | Description                        | Required? |
|-----------------|------------------------------------|-----------|
| AdjudicationUid | The Appeal UID you are escalating. | Yes       |

### Returned XML Parameters

| Parameter Name             | Description  |
|----------------------------|--|
| EscalateAdjudicationResult | Indicates with a 1 or 0 in the results string whether or not the appeal has been escalated to the next workflow stage. |

### Usage Notes

This web method does not use primitive types. Rather, the information is encapsulated in a special wrapper object (complex type). Users will not be able to use the web method from the discovery page but will instead need to use a suitable test program such as *soapUI*.

## InsertAdjudication2() v2.0

The Insert Appeal web method creates new appeal for a citation or boot/tow. InsertAdjudication (without the 2) should be used for Flex versions lower than 7.2.

### Input XML Parameters

| Parameter Name   | Description                      | Required?                   |
|------------------|----------------------------------|-----------------------------|
| ContraventionUid | The citation UID being appealed. | Yes, if a citation is being |

|                           |  |                                      |
|---------------------------|--|--------------------------------------|
|                           |  | appealed                             |
| BootTowUid                | The boot/tow UID being appealed.   | Yes, if a boot/tow is being appealed |
| AdjudicationTypeUid       | The appeal type UID.   | Yes                                  |
| EntityUid                 | Customer UID appealing the citation or boot/tow.   | Yes                                  |
| HoldEscalationTypeUid     | How to handle escalations while on appeal.   | No                                   |
| AddressUid                | The address UID where correspondence for the appeal should be sent.  | No                                   |
| EmailUid                  | The email address UID where correspondence for the appeal should be sent.  | No                                   |
| NoteCodeUid               | The appeal note code to apply when inserting the appeal.   | No                                   |
| AdjudicationDate          | Date of the appeal in the format YYYY-MM-DD.   | Yes                                  |
| UsePDF                    | Use 1/0 values to indicate yes or no for the email format as PDF or standard HTML. This value is determined by the Appeal Type setting in Flex.  | No                                   |
| ReinstateDaysAfter        | The number of days after appeal is resolved to reinstate the item(s).  | Yes                                  |
| NumberPrefix              | A text string for the appeal number prefix.  | No                                   |
| AlternateId               | A text string for any alternate Id for the appeal.   | No                                   |
| ParameterCustomFieldItems | Contains a CustomFieldItem complex type* parameter set for each custom field on the appeal (see next item).  | No                                   |
| CustomFieldItem           | Complex type* collection of the following parameters used to indicate any custom fields on the appeal: <ul style="list-style-type: none"> <li>■ FieldValue (string)</li> <li>■ CustomFieldItem (string)</li> </ul> | No                                   |

### Returned XML Parameters

| Parameter Name            | Description   |
|---------------------------|---|
| InsertAdjudication2Result | Returns the inserted appeal's UID and workflow step in a string to indicate it was successfully inserted. |

### Usage Notes

- \*Complex Type – This type of parameter contains nested parameters, as indicated in the table.
- This web method does not use primitive types. Rather, the information is encapsulated in a special wrapper object (complex type). Users will not be able to use the web method from the discovery page but will instead need to use a suitable test program such as *soapUI*.

## InsertUpdateCustomFlds() v1.0

### Input XML Parameters

| Parameter    | Description                  | Required? |
|--------------|------------------------------|-----------|
| (FIELD_NAME) | Custom Field Name Ex: Cust_1 | Yes       |

### Return XML Parameters

| Parameter Name               | Description   |
|------------------------------|---|
| InsertUpdateCustomFldsResult | Indicates in the results string whether or not the custom field was successfully inserted or updated. |

### Usage Notes

This web method does not use primitive types. Rather, the information is encapsulated in a special wrapper object (complex type). Users will not be able to use the web method from the discovery page but will instead need to use a suitable test program such as *soapUI*.

This method populates a new or updates an existing custom field on one or more appeal records. The parameter name designated by {FIELD\_NAME} is the name of the custom field to be used. The parameter value is the value to be inserted.

More than one field can be passed by including multiple field names among the XML parameters.

## LinkAdjudicationContraventionsBootTows() v1.0

The Link Appeal Citations Boot Tows web method allows multiple object types to be linked to the given Appeal.

### Input XML Parameters

| Parameter Name          | Description  | Required?                            |
|-------------------------|--|--------------------------------------|
| AdjudicationUid         | The Appeal UID to which to link the given Citations/Boot Tows.   | Yes                                  |
| Items                   | Contains an AdjudicationContraventionBootTowItem complex type* parameter set for each item being linked to the appeal (see next item). | Yes (one for each item being linked) |
| AdjudicationContraventi | Complex type* collection of the following parameters   | Yes                                  |

|               |  |  |
|---------------|--|--|
| onBootTowItem | used to link to the appeal: <ul style="list-style-type: none"> <li>■ SourceObjectType (integer)</li> <li>■ SourceObjectUid (integer)</li> <li>■ NoteCodeUid (integer)</li> </ul> |  |
|---------------|--|--|

### Returned XML Parameters

| Parameter Name                               | Description  |
|--|--|
| LinkAdjudicationContraventionsBookTowsResult | Indicates with a 1 or 0 in the results string whether or not the citation(s) and/or boot/tow(s) were linked to the Appeal. |

### Usage Notes

- This web method does not use primitive types. Rather, the information is encapsulated in a special wrapper object (complex type). Users will not be able to use the web method from the discovery page but will instead need to use a suitable test program such as *soapUI*.
- \*Complex Type – This type of parameter contains nested parameters, as indicated in the table.
- When linking a citation or boot/tow to an existing appeal, no rules will be checked. The appeal will follow the original workflow it was assigned when the first item was added.
- If linking any citation or boot/tow generates an error, all work done up to that point will be rolled back and the error will be returned to the caller.

## PayRelatedAdjudicationFees() v1.0

The Pay Related Fees web method allows positive payments to be applied to the balance on a given appeal.

### Input XML Parameters

| Parameter Name   | Description   | Required? |
|------------------|---|-----------|
| AdjudicationUid  | The appeal UID to pay.                                      | Yes       |
| CashdrawerUid    | The web service cash drawer UID for the payment.            | Yes       |
| PaymentMethodUid | The payment method UID.                                     | Yes       |
| Amount           | The amount of the payment in dollars (allows two decimals). | Yes       |

### Returned XML Parameters

| Parameter Name                   | Description   |
|----------------------------------|---|
| PayRelatedAdjudicationFeesResult | Returns the receipt UID in a string to indicate the appeal fee was successfully paid. |

### Usage Notes

This web method does not use primitive types. Rather, the information is encapsulated in a special wrapper object (complex type). Users will not be able to use the web method from the discovery page but will instead need to use a suitable test program such as *soapUI*.

## UpdateAdjudicationNoteCode() v1.0

This Web method is used to update note codes on appeals.

### Input XML Parameters

| Parameter Name                  | Description   | Required?                               |
|---------------------------------|---|---|
| UpdateAdjudicationNoteCodeParam | Complex type* collection of the following parameters used to update each appeal: <ul style="list-style-type: none"> <li>■ AdjudicationUID</li> <li>■ ContraventionUID</li> <li>■ BootTowUID</li> <li>■ NoteCodeUID</li> </ul> | Yes (one for each appeal being updated) |

### Returned XML Parameters

| Parameter Name                   | Description   |
|----------------------------------|---|
| UpdateAdjudicationNoteCodeResult | Returns the UIDs in a string for the appeal, each item on the appeal, and the associated note code(s) to indicate the note codes were successfully updated. |

### Usage Notes

- This web method does not use primitive types. Rather, the information is encapsulated in a special wrapper object (complex type). Users will not be able to use the web method from the discovery page but will instead need to use a suitable test program such as *soapUI*.
- Note codes can be empty, in which case the blank note code will be assigned.
- Users can update multiple note codes at once; however, all updates must be successful. If one fails, the entire call will fail.
- \*Complex Type – This type of parameter contains nested parameters, as indicated in the table.

## T2\_FLEX\_ALLOCATIONS

### AllocationAddEntity() v1.0

#### Input XML Parameters

| Parameter Name | Description  | Required? |
|----------------|--|-----------|
| ALLOCATION_UID | The UID corresponding to the permit allotment being altered. | Yes       |
| ENTITY_UID     | The UID corresponding to the customer being added.           | Yes       |

#### Return XML Parameters

| Parameter Name | Description  |
|----------------|--|
| ENTITY_ADDED   | Returns 1 if the customer was added. The parameter is not returned in the event of an error. |

#### Usage Notes

This web method adds an existing customer to the list of customers on an allotment (called allocation here).

Customers can be added to an allotment even if they are not linked to the proper allotment group. T2 Flex will check customer eligibility to purchase during the permit sales process.

### AllocationGetAllocationList() v1.0

#### Input XML Parameters

| Parameter Name | Description  | Required? |
|----------------|--|-----------|
| ENTITY_UID     | The UID corresponding to the allotment group (department) being queried. | Yes       |

#### Return XML Parameters

| Parameter Name         | Description   |
|------------------------|---|
| PAN_UID                | The permit allotment UID that is linked to the requested allotment group.         |
| PAN_TARGET_DESCRIPTION | The target description (facility or facility group description) of the allotment. |
| ATP_DESCRIPTION        | The time period description for the allotment (Spring, Winter, Fall, etc.)        |
| ATP_START_DATE         | The start date on the time period for the allotment.                              |

|                         |  |
|-------------------------|--|
| ATP_END_DATE            | The end date on the time period for the allotment. |
| PAN_NUM_VALID_SOLD_PERM | The number of valid sold permits on the allotment. |
| PAN_NUM_REMAINING_PERM  | The number of remaining permits on the allotment.  |
| PAN_NUM_ALLOC_PERM      | The number of permits (total) on the allotment.    |

### Usage Notes

This web method provides a list of allotments for a particular allotment group (department).

## AllocationGetEntityList() v1.0

### Input XML Parameters

| Parameter Name | Description  | Required? |
|----------------|--|-----------|
| ALLOCATION_UID | The UID corresponding to the permit allotment being queried. | Yes       |

### Return XML Parameters

| Parameter Name          | Description  |
|-------------------------|--|
| ENT_UID                 | The customer UID corresponding to the customer using the allotment.  |
| ENT_FULL_NAME           | The full name of the customer using the allotment.   |
| ENT_RELATED_GROUP_NAME  | The name of the allotment group / department to which the customer is a member.                                    |
| PWL_UID_WORKFLOW_STATUS | The workflow status UID of the active permit associated to the customer. Blank if no active permit exists.         |
| PSL_UID_STATUS          | The status UID of the active permit associated to the customer. Blank if no active permit exists.                  |
| PWL_DESCRIPTION         | The workflow status description of the active permit associated to the customer. Blank if no active permit exists. |
| PSL_DESCRIPTION         | The status description of the active permit associated to the customer. Blank if no active permit exists.          |
| PER_NUMBER              | The permit number of the active permit associated to the customer.   |

### Usage Notes

This web method returns a list of customers and the current active permit the customer has that uses an allotment slot. Customers with no active permits will be returned with blank permit data.

## AllocationRemoveEntity() v1.0

### Input XML Parameters

| Parameter Name | Description  | Required? |
|----------------|--|-----------|
| ALLOCATION_UID | The UID corresponding to the permit allotment being altered. | Yes       |
| ENTITY_UID     | The UID corresponding to the customer being removed.         | Yes       |

### Return XML Parameters

| Parameter Name | Description  |
|----------------|--|
| ENTITY_REMOVED | Returns 1 if the customer was removed. The parameter is not returned if an error occurs. |

### Usage Notes

This web method removes a customer from the list of customers on a permit allotment.

## AllocationSearchAllocationGroups() v1.0

### Input XML Parameters

| Parameter Name    | Description   | Required?                                 |
|-------------------|---|---|
| ENTITY_UID        | The UID corresponding to the allotment group being queried. | Yes, unless the group name is provided.   |
| ENTITY_GROUP_NAME | The group name  | Yes, unless the customer UID is provided. |

### Return XML Parameters

| Parameter Name    | Description  |
|-------------------|--|
| RPT_ENTITY_VIEW.* | All fields in the view RPT_CUSTOMER_VIEW are returned from this web service. |

### Usage Notes

This web method searches for a group customer using a wildcard (“begins with” search) and returns the information related to the group customer.



## T2\_FLEX\_BULKPERMISSIONS

### CreateBulkPermission() v1.0

This Web method creates a new bulk permit stack that will then be available to sell using the SellBulkPermission() Web method.

#### Input XML Parameters

| Parameter                 | Description  | Required?                             |
|---------------------------|--|---------------------------------------|
| PermissionNumberRangeUid  | UID of the permit number range from which the bulk permit is created. The number range must be non-inventoried permits.  | Yes                                   |
| PermissionControlGroupUid | UID of the control group from which the bulk permit is created.  | Yes                                   |
| EntityUid                 | UID of the responsible customer.   | Yes                                   |
| CashDrawerUid             | UID of the cash drawer selling the permits.  | Yes                                   |
| PermissionUidBegin        | UID of the first permit in the bulk stack.   | Yes, if NumberOfPermits is blank.     |
| PermissionUidEnd          | UID of the last permit in the bulk stack.  | Yes, if NumberOfPermits is blank.     |
| NumberOfPermits           | Number of permits sold to the bulk stack.  | Yes, if begin and end UIDs are blank. |
| EffectiveDate             | Effective date of the bulk permits.  | Yes                                   |
| ExpirationDate            | Expiration date of the bulk permits.   | Yes                                   |
| IsConsigneeResponsible    | 1 or 0, Yes or No indicates whether or not the consignee will be responsible for citations when regular citation assignment rules cannot determine a responsible customer. | Yes                                   |

#### Return XML Parameters

| Parameter                  | Description  |
|----------------------------|--|
| CreateBulkPermissionResult | A string indication the bulk permit was successfully inserted or not, including the BulkPermissionUid and the following information for each permit in the stack:<br>PermissionUid<br>PermissionNumber |

|  |   |
|--|---|
|  | BulkPermissionTax1<br>BulkPermissionTax2<br>BulkPermissionTax3<br>BulkPermissionTotalPrice<br>PermissionCount |
|--|---|

## DeleteBulkPermission() v1.0

This Web method deletes an existing bulk permit previously created by the CreateBulkPermission() Web method.

### Input XML Parameters

| Parameter         | Description             | Required? |
|-------------------|-------------------------|-----------|
| BulkPermissionUid | UID of the bulk permit. | Yes       |
| EntityUID         | UID of the entity.      | Yes       |

### Return XML Parameters

| Parameter                  | Description   |
|----------------------------|---|
| DeleteBulkPermissionResult | A string containing the BulkPermissionUid, indicating the bulk permit was successfully deleted. |

## SellBulkPermission() v1.0

This Web method sells an existing bulk permit previously created by the CreateBulkPermission() Web method.

### Input XML Parameters

| Parameter                  | Description  | Required?                      |
|----------------------------|--|--------------------------------|
| BulkPermissionUid          | UID of the bulk permit.  | Yes                            |
| PaymentMethodUID           | UID of the payment method used for the sale.   | Yes                            |
| EntityUID                  | UID of the purchasing customer.  | Yes                            |
| CashDrawerUID              | UID of the Web drawer.   | Yes                            |
| PayplanStartDate           | Start date of the payment plan, if used as the payment method.                         | Required only for payment plan |
| TransferAgentUID           | UID of the transfer agency, if being paid for with an external payment plan.           | Required only for payment plan |
| PaymentScheduleTemplateUID | UID of the payment schedule template, if being paid for with an external payment plan. | Required only for payment plan |

|                    |                                       |                                |
|--------------------|---------------------------------------|--------------------------------|
| PaymentPlanTaxType | Payment plan tax type.                | Required only for payment plan |
| ReferenceNumber    | Reference number for the transaction. | No                             |

**Return XML Parameters**

| Parameter                | Description   |
|--------------------------|---|
| SellBulkPermissionResult | A string indicating whether or not the bulk permit was successfully sold. |

**T2\_FLEX\_CC\_PADSS**

This Web service and its methods are for internal T2 Systems use only and should not be used.

**T2\_FLEX\_CONTRAVENTIONS****InsertUpdateCustomFlds() v1.0****Input XML Parameters**

| Parameter         | Description                           | Required? |
|-------------------|---------------------------------------|-----------|
| (FIELD_NAME)      | Custom Field Name Ex: Cust_1          | Yes       |
| CONTRAVENTION_UID | Citation UID (CONTRAVENTION.CON_UID). | Yes       |

**Return XML Parameters**

| Parameter         | Description                                    |
|-------------------|--|
| CONTRAVENTION_UID | UID of citation that was successfully updated. |

**Usage Notes**

This method populates an existing custom field on the citation record specified by CONTRAVENTION\_UID. The parameter name designated by {FIELD\_NAME} is the name of the custom field to be used. The parameter value is the value to be inserted.

More than one field can be passed by including multiple field names among the XML parameters.

**PayContravention() v1.3****Input XML Parameters**

| Parameter            | Description   | Required? |
|----------------------|---|-----------|
| CONTRAVENTION_NUMBER | A comma separated list of citation numbers (CONTRAVENTION.CON_TICKET_ID). | Yes       |

|                            |  |  |
|----------------------------|--|--|
| PAYMENT_TYPE               | Payment Method UID (PAYMENT_METHOD_MLKP.PMM_UID).  | Yes  |
| AMOUNT_PAID                | Total amount paid, including any taxes. Format is decimal, e.g., 6.25.   | Yes  |
| REFERENCE_NOTE             | Reference Note.  | No   |
| CASH_DRAWER_UID            | Cash Drawer to use for this transaction (CASHDRAWER_LKP.CDL_UID).  | Yes  |
| PAYPLAN_TRANSFER_AGENT_UID | Payment Plan Transfer Agent UID (TRANSFER_AGENCY_MKLP.TAL_UID).  | Only if a pay method of external pay plan is used.                                   |
| PAYPLAN_TEMPLATE_UID       | Payment Plan Template UID (PAYMENT_SCHEDULE_TEMP_MLKP.PSE_UID).  | No   |
| PAYPLAN_START_DATE         | Payment Plan Start Date. Format is MM/DD/YYYY.   | No, if pay plan used and no date specified the current date will be used.            |
| PAYPLAN_TIME_PERIOD_UID    | Payment plan time period UID (PAYMENT_PLAN_PERIOD_LKP.PPP_UID). Values: 1-Weekly, 2-Semi-Monthly, 3-Monthly                        | Yes, if pay plan used and template not specified.                                    |
| PAYPLAN_TAX_TYPE           | Payment plan tax type UID (PAY_PLAN_TAX_STATUS_LKP.PAT_UID). Values: 2- Pre-Tax, 3- Post-Tax                                       | Yes, if an external pay plan used and required by transfer agent.                    |
| PAYPLAN_NUM_PAYMENTS       | Number of payment plan payments. Must be a whole number.   | Yes, if pay plan used, template not specified, and amount per payment not specified. |
| PAYPLAN_AMOUNT_PER_PAYMENT | Amount of each payment plan payment  | Yes, if pay plan used, template not specified, and number of payments not specified. |
| ENTITY_UID                 | Entity UID (ENTITY.ENT_UID) if the citations are paid by a payment plan or UID of the customer associated with the Account credit. | Yes, if payment plan or Account payment method used.                                 |
| ACCOUNT_CREDIT_LIST        | A comma separated list of Account Credits (ACCOUNT_CREDITS.ACR_UID).   | Yes, if account payment method used  |
| (FIELD_NAME)               | Custom Field Name (Ex: Cust_1)   | No   |

**Return XML Parameters**

| Parameter                                 | Description   |
|---|---|
| CONTRAVENTION_LIST                        | The list of citation numbers that was passed in.  |
| RECEIPT_UID                               | UID of the receipt for this payment.  |
| CONTRAVENTION_LIST_PARTIAL_PAYMENT        | Ticket ID of the citation that was partially paid. This item can be empty.  |
| CONTRAVENTION_LIST_PARTIAL_PAYMENT_AMOUNT | The dollar amount partially paid on the citation listed in the CONTRAVENTION_LIST_PARTIAL_PAYMENT element. If the CONTRAVENTION_LIST_PARTIAL_PAYMENT element is empty this value will be 0. If the CONTRAVENTION_LIST_PARTIAL_PAYMENT element is not empty this item will always be greater than 0. |
| CONTRAVENTION_LIST_NO_PAYMENT             | Citation Ticket ID's which were not paid in full or partially paid during the transaction. This item can be empty.  |

**Usage Notes**

This method is used to mark one or more citations as having been paid.

Multiple citations can be passed with a comma delimited string (e.g., ABC0001,ABC0002). If multiple citation numbers are passed, the payment amount passed should equal the balance due for all of the citations, including any taxes. If an error occurs with paying one of the citations, an error will be returned and none of the citations will be marked as paid.

Partial payments and overpayments are allowed if the amount is within the Web services cash drawer user limits.

**Note:** When partially paying citations, they will be paid in the order supplied in the CONTRAVENTION\_LIST. This leaves only one citation partially paid. Any remaining citations will still have a full balance due. When paying multiple citations the overpayment will only be applied to the last citation in the basket.

This method cannot use the following payment types:

- Import Payment
- Payment Plan Refund
- Third Party

Note the following properties of citation payments using an internal or external payment plan for the payment type:

- The transfer agency is only required for external payment plans, but any associated custom fields are not required.

- For external payment plans, if a payment plan template UID is specified, the payment plan time period, and number of payments or amount per payment are not required. However, if a payment plan template is specified, then any values passed for payment plan time period, number of payments, or the amount per payment will override the values found in the template. Therefore, to use the values on a payment plan template do not send the PAYPLAN\_TIME\_PERIOD\_UID, PAYPLAN\_NUM\_PAYMENTS, or the PAYPLAN\_AMOUNT\_PER\_PAYMENT parameters.
- If a valid payment plan template is not specified, the payment plan time period and either the number of payment or the amount per payment are required. If both the number of payments and the amount per payment are included an error will be returned. Note: A valid payment plan template must have an UID greater than zero (0).
- If the number of payments exceeds the expiration date of the permission, an error will not be returned.
 

Note: In T2 Flex this is an overrideable error, however the Web service will allow it to occur without a warning.
- The payment plan tax type is only required if the “Set Pre/Post Tax at Checkout” flag is selected on the Transfer Agency setup. If that flag is not set any value sent for payment plan tax type will not be used.
- If the number of payments is specified then the amount per payment will be calculated and is not required. If the amount per payment is specified then the number of payments will be calculated and is not required. If both the number of payments and the amount per payment are specified then an error will be returned.

The ENTITY\_UID parameter is used only in conjunction with a payment plan or account payment method.

## PrePayContraventions() v1.0

### Input XML Parameters

| Parameter            | Description                                      | Required? |
|----------------------|--|-----------|
| CONTRAVENTION_NUMBER | The number on the ticket.                        | Yes       |
| VIOLATION_UID        | Violation UID (CON_VIOLATION_CODE_MLKP.VIC_UID). | Yes       |
| LICENSE_PLATE        | License plate number                             | No        |
| STATE_UID            | State UID (STATE_MLKP.STL_UID)                   | No        |
| FULL_NAME            | Name of person making the payment                | No        |
| PAYMENT_TYPE         | Payment Method UID                               | Yes       |

|                 |  |     |
|-----------------|--|-----|
|                 | (PAYMENT_METHOD_MLKP.PMM_UID)  |     |
| AMOUNT_PAID     | Total amount paid, including any taxes.<br>Format is decimal, e.g., 6.25 | Yes |
| REFERENCE_NOTE  | Reference note   | No  |
| CASH_DRAWER_UID | Cash Drawer to use for this transaction<br>(CASHDRAWER_LKP.CDL_UID)      | Yes |

**Return XML Parameters**

| Parameter            | Description                             |
|----------------------|---|
| CONTRAVENTION_NUMBER | The citation number that was passed in. |
| RECEIPT_UID          | UID of the receipt for this payment     |

**Usage Notes**

Only one citation can be pre-paid at a time.

Pre-paid citations cannot be put on a payment plan or be paid by a third party.

**RecallContravention() v1.0**

This Web method recalls a citation that has been hard-transferred so that it can be appealed.

**Input XML Parameters**

| Parameter   | Description                         | Required? |
|-------------|-------------------------------------|-----------|
| CitationUid | UID of the citation to be recalled. | Yes       |

**Return XML Parameters**

| Parameter                 | Description  |
|---------------------------|--|
| RecallContraventionResult | A string indicating the citation was successfully recalled or not. |

**TransferContravention() v1.0**

This Web method transfers a regular or transfer- pending citation completely (bypassing the citation transfer pending process in Flex) so that it can be recalled for appeal using the RecallContravention() Web method.

Note: This web method is used solely for the purpose of allowing recall of a citation. The Citation batch task in Flex will not pick up citations that have been transferred via Web Services.

**Input XML Parameters**

| Parameter         | Description                            | Required? |
|-------------------|--|-----------|
| ContraventionUid  | UID of the citation to be transferred. | Yes       |
| TransferAgencyUid | UID of the transfer agency.            | Yes       |

**Return XML Parameters**

| Parameter                   | Description   |
|-----------------------------|---|
| TransferContraventionResult | A string indicating the citation was successfully transferred or not. |

**UpdateContraventionSpecialStatus() v1.0****Input XML Parameters**

| Parameter         | Description | Required? |
|-------------------|-------------|-----------|
| CONTRAVENTION_UID | Integer     | Yes       |
| IS_SPECIAL_STATUS | Boolean     | Yes       |

**Return XML Parameters**

| Parameter         | Description                   |
|-------------------|-------------------------------|
| CONTRAVENTION_UID | Will match the submitted UID. |

**Usage Notes**

This method is used to update the Special Status field on a citation to either true or false.

**T2\_FLEX\_DOCKETS****ScheduleAdjudication() v1.0**

The Schedule Appeal web method will allow an appeal at the correct workflow stage to be scheduled on a docket. The docket must be open and within the date range allowed by System Settings. Any officer on a citation or boot/tow linked to the appeal must also appear on the docket if issuing officers are required. We do not just test a “primary” item, but all of them.

Any issuing officers required by the docket must be the same as the primary officer on the citation or boot/tow.

**Input XML Parameters**

| Parameter Name  | Description                                | Required? |
|-----------------|--|-----------|
| AdjudicationUid | UID of the Appeal to schedule.             | Yes       |
| DocketUid       | UID of the Docket on which to schedule the | Yes       |



|  |         |  |
|--|---------|--|
|  | Appeal. |  |
|--|---------|--|

### Returned XML Parameters

| Parameter Name             | Description  |
|----------------------------|--|
| ScheduleAdjudicationResult | Indicates with a 1 or a 0 in the results string whether or not the appeal was scheduled on the docket. |

### Usage Notes

This web method does not use primitive types. Rather, the information is encapsulated in a special wrapper object (complex type). Users will not be able to use the web method from the discovery page but will instead need to use a suitable test program such as *soapUI*.

## T2\_FLEX\_ENCODINGMODULE

This Web service and its methods are for internal T2 Systems use only and should not be used.

## T2\_FLEX\_ENTITIES

### CreditAccount() v1.0

#### Input XML Parameters

| Parameter             | Description  | Required? |
|-----------------------|--|-----------|
| ENTITY_UID            | Customer UID (ENTITY.ENT_UID) of the customer associated with the Account to credit. | Yes       |
| CASH_DRAWER_UID       | Cash Drawer to use for this transaction (CASHDRAWER_LKP.CDL_UID).                    | Yes       |
| CREDIT_AMOUNT         | The dollar amount of the Account Credit to create.                                   | Yes       |
| CREDIT_PAYMENT_METHOD | The Payment Method to use for the Account Credit.                                    | Yes       |
| GL_ACCOUNT_UID        | The GL Account to associate with the dollar amount.                                  | Yes       |
| CONSUME_AUTOMATICALLY | Used to determine if the Account Credit can be consumed automatically.               | Yes       |
| REFERENCE_NOTE        | Reference note.  | No        |

**Return XML Parameters**

| Parameter          | Description                                  |
|--------------------|--|
| ACCOUNT_CREDIT_UID | The UID of the created AccountCredit object. |

**Usage Notes**

This method allows a dollar amount to be refunded to a customer account balance for a specified payment method. It can only be used when the system is configured for MCI.

This method only supports payment methods configured to be valid for a Deposit to Account transaction.

**InsertUpdateAddress() v1.0****Input XML Parameters**

| Parameter        | Description   | Required                             |
|------------------|---|--------------------------------------|
| ADDRESS_UID      | Address UID (COR_ADDRESS.COR_UID). This parameter is only sent if the address is being updated.   | No, if inserting<br>Yes, if updating |
| ADDRESS_TYPE     | Address type code (COR_ADD_TYPE_LKP.CAL_UID).   | Yes, if inserting<br>No, if updating |
| PRIMARY_STREET   | Primary Street.   | Yes, if inserting<br>No, if updating |
| SECONDARY_STREET | Secondary Street.   | No                                   |
| SUITE_APT        | Suite or Apartment.   | No                                   |
| CITY             | City.   | No                                   |
| STATE_UID        | State UID (STATE_MLKP.STL_UID)  | No                                   |
| COUNTRY          | Country (COUNTRY_LKP.COL_UID).  | No                                   |
| POSTAL_CODE      | Postal Code.  | No                                   |
| ADDRESS_PRIORITY | Address priority. Format is integer, with 1 representing highest priority relative to other addresses for this entity. Default value is 1 for insert. | No                                   |
| START_DATE       | Date address became/becomes active. Format: MM/DD/YY  | Yes, if inserting                    |
| END_DATE         | Date address became/becomes inactive. Format: MM/DD/YY  | No                                   |
| ENTITY_UID       | Customer UID to which the address will be added (ENTITY.ENT_UID). Do not pass   | Yes, if inserting                    |

|                    |   |    |
|--------------------|---|----|
|                    | this parameter for an update. This parameter cannot be used to update an address with a different customer UID. |    |
| AUTO_LINK_PROPERTY | Flag indicating if a property should be automatically linked to an entity.                                      | No |

### Return XML Parameters

| Parameter   | Description  |
|-------------|--|
| ADDRESS_UID | UID of address that was successfully inserted or updated |

### Usage Notes

This method is used to insert a new address or update an existing address for an entity.

If an address UID is not passed, then the method will attempt to insert a new address using the information passed into the method.

If an address UID is passed, then method will attempt to update an existing address using information passed into the method.

If an address priority is passed then the method will reprioritize all addresses for the customer in order to place the address at the specified priority position. If a priority greater than that of any existing address is passed, then the address should be moved to the last position.

If no address priority is passed on an insert, the address will be inserted at priority position 1. If no address priority is passed on an update, the priority of the address will be unchanged.

Note that the fields comprising the address (street, city, country, etc.) are all text fields, with the exception of state, for which a database UID must be passed into the STATE\_UID parameter.

## InsertUpdateCustomFlds() v1.0

### Input XML Parameters

| Parameter    | Description                    | Required |
|--------------|--------------------------------|----------|
| (FIELD_NAME) | Custom Field Name Ex: Cust_1   | Yes      |
| ENTITY_UID   | Customer UID (ENTITY.ENT_UID). | Yes      |

### Return XML Parameters

| Parameter  | Description                                    |
|------------|--|
| ENTITY_UID | UID of customer that was successfully updated. |

### Usage Notes

This method populates an existing custom field on the customer record specified by ENTITY\_UID. The parameter name designated by {FIELD\_NAME} is the name of the custom field to be used. The parameter value is the value to be inserted.

More than one field can be passed by including multiple field names among the XML parameters.

## InsertUpdateEmailAddress() v1.0

### Input XML Parameters

| Parameter        | Description  | Required                             |
|------------------|--|--------------------------------------|
| EMAIL_UID        | Email UID (COR_EMAIL.COE_UID).   | No, if inserting<br>Yes, if updating |
| EMAIL_TYPE       | Email type code (COE_ADD_TYPE_LKP.CEL_UID).  | Yes, if inserting<br>No, if updating |
| EMAIL_ADDRESS    | Email Address.   | Yes                                  |
| ADDRESS_PRIORITY | Address priority. Format is integer, with 1 representing highest priority relative to other addresses for this entity. Default value is 1 for insert.  | No                                   |
| START_DATE       | Date address became/becomes active. Format: MM/DD/YY.  | Yes, if inserting                    |
| END_DATE         | Date address became/becomes inactive. Format: MM/DD/YY.  | No                                   |
| ENTITY_UID       | Customer UID (ENTITY.ENT_UID) to which the address will be added. Do not pass this parameter for an update. This parameter cannot be used to update an e-mail address with a different customer UID. | Yes                                  |

### Return XML Parameters

| Parameter | Description   |
|-----------|---|
| EMAIL_UID | UID of email address that was successfully inserted or updated. |

### Usage Notes

This method is used to insert a new email address or update an existing email address for an entity.

If an email UID is not passed, then the method will attempt to insert a new email address using the information passed into the method.

If an email UID is passed, then method will attempt to update an existing email address using information passed into the method.

If an address priority is passed then the method will reprioritize all email addresses for the customer in order to place the address at the specified priority position. If a priority greater than that of any existing email address is passed, then the email address should be moved to the last position.

If no address priority is passed on an insert, the email address will be inserted at priority position 1. If no address priority is passed on an update, the priority of the email address will be unchanged.

## InsertUpdateEntity() v1.0

### Input XML Parameters

| Parameter             | Description  | Required                             |
|-----------------------|--|--------------------------------------|
| ENTITY_UID            | Entity UID (ENTITY.ENT_UID). This parameter is only sent if the customer is being updated. | No, if inserting<br>Yes, if updating |
| ENTITY_TYPE           | Entity type code (ENT_TYPE_LKP.ETL_UID)  | Yes, if inserting<br>No, if updating |
| GROUP_NAME            | Group name   | No, unless entity_type is Group      |
| PRIMARY_UID           | Primary UID  | No                                   |
| SECONDARY_UID         | Secondary UID  | No                                   |
| TERTIARY_UID          | Tertiary UID   | No                                   |
| FIRST_NAME            | First Name   | No                                   |
| MIDDLE_NAME           | Middle Name  | No                                   |
| LAST_NAME             | Last Name  | No                                   |
| NAME_PREFIX_UID       | Prefix for the entity's name (ENT_NAME_PREFIX_LKP.ENP_UID)                                 | No                                   |
| NAME_SUFFIX_UID       | Suffix for the entity's name (ENT_NAME_SUFFIX_LKP.ENS_UID)                                 | No                                   |
| SUB_CLASSIFICATION    | Sub-Classification code (ENT_SUBCLASSIFICATION_MLKP.ESL_UID)                               | Yes, if inserting                    |
| PHONE_ONE             | First phone number   | No                                   |
| PHONE_TWO             | Second phone number  | No                                   |
| PHONE_THREE           | Third phone number   | No                                   |
| ENT_UID_RELATED_GROUP | The UID corresponding to the customer to set as the Allotment Group.                       | No                                   |

|              |                                |    |
|--------------|--------------------------------|----|
| (FIELD_NAME) | Custom Field Name (Ex: Cust_1) | No |
|--------------|--------------------------------|----|

**Return XML Parameters**

| Parameter  | Description   |
|------------|---|
| ENTITY_UID | UID of customer that was successfully inserted or updated |

**Usage Notes**

This method is used to insert a new customer record or update an existing customer record.

- If a customer UID is not passed in then the method will attempt to insert a new customer using the information passed into the method.
- If a customer UID is passed in then method will attempt to update an existing customer using the information passed into the method.
- It is possible, although not advisable, to insert a customer having no identifying information, as the various name and UID fields are not required for an insert.
- A Related Group may be removed on an edit by supplying a space (“ ”) as the value of the parameter.

**InsertUpdatePropertyRelationship() v1.0****Input XML Parameters**

| Parameter             | Description                         | Required? |
|-----------------------|-------------------------------------|-----------|
| RELATIONSHIP_UID      | Relationship to update.             | No        |
| ENTITY_UID            | The entity.                         | Yes       |
| PROPERTY_UID          | The property.                       | Yes       |
| RELATIONSHIP_TYPE_UID | The type of relationship.           | Yes       |
| START_DATE            | The start date of the relationship. | Yes       |
| END_DATE              | The end date of the relationship.   | No        |

**Return XML Parameters**

| Parameter        | Description                                 |
|------------------|---|
| RELATIONSHIP_UID | The UID of the new or updated relationship. |

**Usage Notes**

This method allows properties to be linked to customers. This method can also be used to update an existing relationship.

**PayOutAccount() v1.0****Input XML Parameters**

| Parameter Name             | Description  | Required?  |
|----------------------------|--|--|
| ACCOUNT_CREDIT_LIST        | Used to provide a list of Account Credits to use for payments.                               | Yes if Account payment method used                                       |
| (FIELD_NAME)               | Custom Field Name (Ex: Cust_1)   | No   |
| PAYMENT_TYPE               | Payment Method UID (PAYMENT_METHOD_MLKP.PMM_UID).  | Yes  |
| AMOUNT_PAID                | Total to pay out.  | Yes, if MCI is disabled.   |
| REFERENCE_NOTE             | Reference note.  | No   |
| CASH_DRAWER_UID            | Cash Drawer to use for this transaction (CASHDRAWER_LKP.CDL_UID).                            | Yes  |
| PAYPLAN_TRANSFER_AGENT_UID | Payment Plan Transfer Agent UID (TRANSFER_AGENCY_MKLP.TAL_UID).                              | Yes, if a pay method of external pay plan is used.                       |
| PAYPLAN_START_DATE         | Payment Plan Start Date. Format is MM/DD/YYYY.   | No, if pay plan used and no date specified the current date will be used |
| PAYPLAN_TAX_TYPE           | Payment plan tax type UID (PAY_PLAN_TAX_STATUS_LKP.PAT_UID). Values: 2- Pre-Tax, 3- Post-Tax | Yes, if an external pay plan used and                                    |
| ENTITY_UID                 | Customer UID (ENTITY.ENT_UID) associated with the account being paid out.                    | Yes, if pay plan used.   |

**Return XML Parameters**

| Parameter Name | Description                     |
|----------------|---------------------------------|
| RECEIPT_UID    | The UID of the created receipt. |

### Usage Notes

This method allows individual account credits to be paid out using any of the configured payout methods. It does not support the following payment types:

- Account Deposit
- Import Payment
- Payment Plan Refund
- Third Party

## T2\_FLEX\_EVENTS

### ApproveEvent() v1.0

This Web method allows an event to be approved.

#### Input XML Parameters

| Parameter | Description                  | Required |
|-----------|------------------------------|----------|
| EventUid  | UID of event to be approved. | Yes      |

#### Return XML Parameters

| Parameter          | Description  |
|--------------------|--|
| ApproveEventResult | A string describing whether or not the event was approved. |

### FinalizeEvent() v1.0

This Web method allows an event to be finalized.

#### Input XML Parameters

| Parameter | Description                   | Required |
|-----------|-------------------------------|----------|
| EventUid  | UID of event to be finalized. | Yes      |

#### Return XML Parameters

| Parameter           | Description   |
|---------------------|---|
| FinalizeEventResult | A string describing whether or not the event was finalized. |

### GetEventFacilityList() v1.0

This web method retrieves event facilities, event value and capacity for a given event. If data from GetEventList is cached, running this web method may be unnecessary.



**Input XML Parameters**

| Parameter                     | Description   | Required |
|-------------------------------|---|----------|
| EventUID                      | UID of the event for which a list of event facilities should be returned. | Yes      |
| SearchDate                    | Events that are active or begin after this date are returned.             | No       |
| CutoffDate                    | Events that begin after this date are not returned.                       | No       |
| ApplyDateFilterToCapacityList | If true, only capacity between the search and cutoff dates is returned    | No       |

**Return XML Parameters**

| Parameter                  | Description  |
|----------------------------|--|
| GetEventFacilityListResult | A string of all available event facilities, event value, and capacity for a given event. |

**Usage Notes**

If both the SearchDate and CutoffDate parameters are omitted, all event facilities for the event are returned even if they have already occurred.

**GetEventList() v1.0**

This web method retrieves a list of events, event facilities and associated event value for sale.

**Input XML Parameters**

| Parameter                          | Description  | Required |
|------------------------------------|--|----------|
| SearchDate                         | Events that are active or begin after this date are returned.                                | No       |
| CutoffDate                         | Events that begin after this date are not returned.  | No       |
| EventValueCanBePurchasedOnly       | If true, only event value that can be purchased (capacity is not sold out) will be returned. | No       |
| ApplyDateFilterToEventFacilityList | If true, only event facilities between the search and cutoff dates are returned.             | No       |
| ApplyDateFilterToCapacityList      | If true, only capacity between the search and cutoff dates is returned                       | No       |

**Return XML Parameters**

| Parameter          | Description  |
|--------------------|--|
| GetEventListResult | A string of all available events, event facilities, event value, and capacity. |

**Usage Notes**

If both the SearchDate and CutoffDate parameters are omitted, all events in the system are returned even if they have already occurred.

**GetEventValueCapacity() v1.0**

This web method gets event value capacity data for a given event facility.

**Input XML Parameters**

| Parameter        | Description  | Required |
|------------------|--|----------|
| EventFacilityUid | UID of the event facility for which the event value capacity should be returned. | Yes      |

**Return XML Parameters**

| Parameter                   | Description  |
|-----------------------------|--|
| GetEventValueCapacityResult | A string of all event value capacity for a given event facility. |

**InsertEvent() v1.0****Input XML Parameters**

| Parameter Name             | Description                                     | Required? |
|----------------------------|---|-----------|
| EVENT_TITLE                | 32 character limit                              | Yes       |
| EVENT_CLASSIFICATION       | UID for an existing Event Classification        | Yes       |
| EVENT_START_DATE           | Format date as MM/DD/YYYY                       | Yes       |
| EVENT_START_TIME           | Format time according to 24 hour clock as HH:MM | Yes       |
| EVENT_END_DATE             | Format date as MM/DD/YYYY                       | Yes       |
| EVENT_END_TIME             | Format time according to 24 hour clock as HH:MM | Yes       |
| EVENT_DESCRIPTION          | 2000 character limit                            | No        |
| EVENT_ESTIMATED_ATTENDANCE | Integer value                                   | No        |

|                                   |   |    |
|-----------------------------------|---|----|
| EVENT_ACTUAL_ATTENDANCE           | Integer value   | No |
| EVENT_DISCOUNT                    | Dollar amount of discount                                       | No |
| APPLY_TAX_1                       | Use 1 or 0 to determine the application of tax                  | No |
| APPLY_TAX_2                       | Use 1 or 0 to determine the application of tax                  | No |
| APPLY_TAX_3                       | Use 1 or 0 to determine the application of tax                  | No |
| EVENT_CONTRAVENTION_LOCATION_LIST | A comma separated list of CLM_UIDs (CON_LOCATION_MLKP.CLM_UID). | No |
| (FIELD_NAME)                      | Custom Field Name (Ex: Cust_1)                                  | No |

### Return XML Parameters

| Parameter Name | Description                         |
|----------------|-------------------------------------|
| EVENT_UID      | The UID of the newly created Event. |

### Usage Notes

Allows you to insert an Event into T2 Flex from the Web, but not any related event records.

- After the event is inserted, the Activity History will read “Event Inserted from WebService.”
- If Events are configured to automatically be approved, the Event will have its status set to Approved.

## InsertEventValue() v1.0

This web method is used to add presold event value to an event facility. Once the event value is added call LinkEventValueToEventCapacity() to link the event value to a capacity record.

This web method is not currently used by e-Business but might be used by a private operator (for example) to allow customers to configure their events hosted at the private operator’s facility.

### Input XML Parameters

| Parameter        | Description   | Required |
|------------------|---|----------|
| EventFacilityUid | UID of the event facility the event value will be added to. | Yes      |

|                              |  |     |
|------------------------------|--|-----|
| PresoldDescription           | Description of the event value that displays when purchasing from it.                        | Yes |
| Cost                         | Cost of the event value.   | Yes |
| NumberOfUses                 | Number of daily entries allowed for the permit with event value.                             | Yes |
| WebAvailableForSaleStartDate | Date event value is available for purchase.  | Yes |
| WebAvailableForSaleEndDate   | Date event value is no longer available for purchase.  | Yes |
| WebTransactionLimit          | Number of permits allowed per online transaction.  | Yes |
| ReportTemplateUid            | Report template UID that will be printed whenever event value is sold.                       | Yes |
| PresoldSalesLimit            | Total number of presold event value permits that can be sold for any given day of the event. | Yes |
| IgnoreWebSalePriority        | If <b>Yes</b> , customers may choose from any available event value when purchasing online.  | Yes |

### Return XML Parameters

| Parameter              | Description   |
|------------------------|---|
| InsertEventValueResult | A string describing the details of the successfully inserted event value. |

## InsertUpdateEntityRelationship() v1.0

This Web method is used to insert or update a relationship between an event and a customer. If the relationship between the event and entity already exists, the contact type is modified to be whatever is passed in as an input. If it does not exist, then the relationship is created with the given contact type.

### Input XML Parameters

| Parameter            | Description                      | Required |
|----------------------|----------------------------------|----------|
| EventUid             | UID of event.                    | Yes      |
| EntityUid            | UID of customer.                 | Yes      |
| EntityContactTypeUid | From<br>EVENT_CONTACT_TYPE_MLKP. | Yes      |

### Return XML Parameters

| Parameter                            | Description  |
|--------------------------------------|--|
| InsertUpdateEntityRelationshipResult | A string describing the customer/event relationship. |

## InsertUpdateEventNote() v1.0

This Web method inserts or updates a note on an event.

### Input XML Parameters

| Parameter       | Description  | Required |
|-----------------|--|----------|
| EventUid        | UID of the event to which the note should be attached.                       | Yes      |
| NoteUid         | UID of note to update. If left blank, the Web method will insert a new note. | No       |
| NoteTypeUid     | UID of note type for inserted/updated note.                                  | No       |
| NoteDescription | Description of inserted/updated note.  | No       |
| NoteEndDate     | End date of inserted/updated note.   | No       |

### Return XML Parameters

| Parameter                   | Description                         |
|-----------------------------|-------------------------------------|
| InsertUpdateEventNoteResult | A string describing the event note. |

## InsertUpdateEventResource() v1.0

This Web method is used to insert or update an event resource linked to an event.

### Input XML Parameters

| Parameter          | Description  | Required                                |
|--------------------|--|---|
| CurrentResourceUid | Matches a known UID in EVENT_REQUIRED_RESOURCES, if blank then this is a new resource entry. | No                                      |
| ResourceTypeUid    | UID of the resource type from EVENT_RESOURCE_MLKP.   | Yes, if CurrentResourceUid is included. |
| EventUid           | UID of the event the resource is linked to.  | Yes, if CurrentResourceUid is included. |
| Quantity           | Quantity needed of the resource.   | No                                      |
| Cost               | Internal information-only cost of the resource.  | No                                      |
| Price              | Price of the resource (per price designation below).   | No                                      |
| PriceType          | 1 = Flat Rate, 2 = Daily Rate, 3 = Hourly Rate   | Yes, if Price is included.              |
| StartDateTime      | Start date/time the resource is needed.  | Yes, if CurrentResourceUid              |

|                   |  |   |
|-------------------|--|---|
|                   |  | is included.                            |
| EndTime           | End date/time the resource is needed.  | Yes, if CurrentResourceUid is included. |
| CheckForConflicts | If True, the web method checks for booking conflicts and returns an error if one is found. | No                                      |

**Return XML Parameters**

| Parameter                       | Description  |
|---------------------------------|--|
| InsertUpdateEventResourceResult | A string describing the inserted/updated event resource. |

**LinkEventValueToEventCapacity() v1.0**

This Web method returns an event value capacity record for a specific day and links it to the newly inserted event value (if no event value capacity exists for the event value).

**Input XML Parameters**

| Parameter             | Description  | Required |
|-----------------------|--|----------|
| EventFacilityUid      | UID of the related event facility.   | Yes      |
| EventValueUid         | UID of the event value to be linked to the capacity schedule.  | Yes      |
| EventValueCapacityUid | UID of the capacity schedule to link. If left blank, a new capacity record is created with a date "DayOfEvent" and maximum capacity "MaxCapacity." | No       |
| DayOfEvent            | Date the event value capacity schedule should be created for.  | Yes      |
| MaxCapacity           | Maximum capacity of the event value capacity for that day.   | No       |

**Return XML Parameters**

| Parameter                           | Description  |
|-------------------------------------|--|
| LinkEventValueToEventCapacityResult | A string containing the newly created/linked event value capacity. |

**PurchaseEventValue() v1.0**

This web method allows event value to be purchased online.

**Input XML Parameters**

| Parameter     | Description                         | Required |
|---------------|-------------------------------------|----------|
| EventValueUid | UID of event value to be purchased. | No       |

|                           |  |  |
|---------------------------|--|--|
| PermissionUid             | UID of permit to which the event value is attached.  | Yes  |
| EntityUid                 | UID of customer that should be linked to the receipt generated by the purchase.  | No   |
| CashDrawerUid             | UID of cash drawer to process the sale.  | Yes  |
| PaymentMethodUid          | UID of the payment method used.  | Yes  |
| AmountPaid                | Amount paid for the event value purchase.  | Yes  |
| MakePurchase              | If <b>False</b> , the purchase will not be made, but the amount due will be calculated.  | No   |
| DatesToPurchase           | Contains a dateTime complex type* parameter set for each event date for which to purchase event value (see next item).                   | Yes, if purchasing by date.                        |
| dateTime                  | Complex type* collection of Date/Time parameters used to indicate each date/time of the event for which event value should be purchased. | Yes, if purchasing by date.                        |
| EventValueItemsToPurchase | Contains an int complex type* parameter set for each event value item that should be purchased (see next item).                          | Yes, if purchasing a specific list of event value. |
| int                       | Complex type* collection of int parameters used to indicate the UID of each event value item that should be purchased.                   | Yes, if purchasing a specific list of event value. |

### Return XML Parameters

| Parameter                | Description  |
|--------------------------|--|
| PurchaseEventValueResult | A string of purchased event value items with receipts if MakePurchase is <b>True</b> . |

### Usage Notes

\*Complex Type – This type of parameter contains nested parameters, as indicated in the table.

## ReserveEventValue() v1.0

This web method reserves event value for sale. Once reserved, the event value is committed unless unreserved or the reservation times out.

### Input XML Parameters

| Parameter     | Description                            | Required |
|---------------|--|----------|
| EventValueUid | UID of the event value to be reserved. | Yes      |

|                  |  |     |
|------------------|--|-----|
| PermissionUid    | UID of the permit to which the event value is attached. If not included, the event value will be sold to a limited-use printable permit. | No  |
| EventDates       | Contains a dateTime complex type* parameter set for each event date for which to reserve event value (see next item).                    | Yes |
| dateTime         | Complex type* collection of Date/Time parameters used to indicate each date/time of the event for which event value should be reserved.  | Yes |
| MinutesToReserve | The web method reserves the event value item for the number of minutes specified. If not specified, the default is 15 minutes.           | No  |

### Return XML Parameters

| Parameter               | Description                                   |
|-------------------------|---|
| ReserveEventValueResult | A string describing the event value reserved. |

### Usage Notes

\*Complex Type – This type of parameter contains nested parameters, as indicated in the table.

During peak sales periods, users could conceivably reserve event value and abandon their sale, leaving potentially many “held” event value spaces that cannot be sold until the reservation expires. Decreasing the MinutesToReserve time will help with this but it shouldn’t be set so low as to expire before a sale can be completed.

## UnreserveEventValue() v1.0

This Web method unreserves event value no longer needed, either by date or by a list of event value item UUIDs. This releases capacity for other sales before the reservation expires. This would be useful during peak sale times to free up reserved event value that is no longer being purchased.

### Input XML Parameters

| Parameter        | Description   | Required                     |
|------------------|---|------------------------------|
| PermissionUid    | UID of the permit to which the event value is attached. | Yes, if unreserving by date. |
| EventValueUid    | UID of the event value to be unreserved.                | Yes, if unreserving by date. |
| DatesToUnreserve | Contains a dateTime complex type*                       | Yes, if                      |



|                            |   |   |
|----------------------------|---|---|
|                            | parameter set for each event date for which to unreserve event value (see next item).   | unreserving by date.                              |
| dateTime                   | Complex type* collection of Date/Time parameters used to indicate each date/time of the event for which event value should be unreserved. | Yes, if unreserving by date.                      |
| EventValueItemsToUnreserve | Contains an int complex type* parameter set for each event value item that should be unreserved (see next item).                          | Yes, if unreserving specific list of event value. |
| int                        | Complex type* collection of int parameters used to indicate the UID of each event value item that should be unreserved.                   | Yes, if unreserving specific list of event value. |

### Return XML Parameters

| Parameter                 | Description                               |
|---------------------------|---|
| UnreserveEventValueResult | A string of unreserved event value items. |

### Usage Notes

\*Complex Type – This type of parameter contains nested parameters, as indicated in the table.

## UpdateEvent() v1.0

This Web method allows editing of custom fields on the event.

### Input XML Parameters

| Parameter        | Description                          | Required |
|------------------|--------------------------------------|----------|
| EventUid         | UID of the event to update.          | Yes      |
| CustomFieldName  | Name of the custom field to update.  | Yes      |
| CustomFieldValue | Value of the custom field to update. | Yes      |

### Return XML Parameters

| Parameter         | Description                      |
|-------------------|----------------------------------|
| UpdateEventResult | A string describing the updates. |

## T2\_FLEX\_HCS

The HCS Alternative Web service checks the application name specified when the handheld connects to the Web service.

- If the application name is PermitNow (only for Flex versions lower than 7.3), the Web service locates the files using the path configured for PermitNow (T2 Flex >

Configuration > System Settings > PermitNow Event > PermitNow HCS Base Directory).

- Otherwise, it uses the path configured for the Handheld Exporter (T2 Flex > Configuration > System Settings > Handheld Exporter > HCS Base Directory Path).

## DownloadHandheldFile() v1.0

### Input XML Parameters

| Parameter Name | Description   | Required? |
|----------------|---|-----------|
| USERNAME       | A username within Flex authorized to perform HCS (same concept as for existing web services, could even use the same username)                            | Yes       |
| PASSWORD       | Password for the specified username   | Yes       |
| VERSION        | As for existing web services  | Yes       |
| FILENAME       | Name of a file being requested e.g. "Events.txt", "Events.zip"  | Yes       |
| START          | First byte position (0 based) of data block. Defaults to 0 if not provided.   |           |
| SIZE           | Size of data block (-1 for entire file (unlimited data block)). Defaults to -1 if not provided.   |           |
| TERMINAL_ID    | Handheld Terminal ID e.g. "HG1", "RECON0001", etc. This could be used by the server to adjust/filter the data sent to the handheld (possible future use). |           |
| WIRELESS_GROUP | Handheld Wireless Group   |           |

### Output XML Parameters

| Parameter | Description  |
|-----------|--|
| FILENAME  | Name of file (should exactly match FILENAME Request Parameter) |
| DATA      | Base64-encoded contents of file                                |
| FILESIZE  | Total size of the file   |
| START     | First byte position (0 based) of data block                    |
| SIZE      | Size of the current data block                                 |

### Usage Notes

The web service will retrieve the file to be downloaded from the folder which serves as the target for the Flex handheld exporter.

The response is a typical HTTP-style response. The response body contains the file contents exactly as found in the LookupFiles folder.

## GetCurrentTime() v1.0

### Input Parameters

| Parameter Name | Description                  | Required? |
|----------------|------------------------------|-----------|
| VERSION        | As for existing Web services | Yes       |

### Output Parameters

| Parameter Name | Description                                |
|----------------|--|
| DATETIME       | Current Date/Time in YYYYMMDDHHMMSS format |

### Usage Notes

This method returns the current Date/Time in YYYYMMDDHHMMSS format. No XMLDATA, USERNAME, or PASSWORD parameters are needed.

## UploadHandheldFile() v1.0

### Input Parameters

| Parameter Name | Description   | Required? |
|----------------|---|-----------|
| USERNAME       | A username within Flex authorized to perform HCS (same concept as for existing web services, could even use the same username)  | Yes       |
| PASSWORD       | Password for the specified username   | Yes       |
| TERMINAL_ID    | A terminal ID which the web server could use at its option when formulating unique filenames for storing uploaded files. This information can be ignored if not required. | No        |
| FILENAME       | Name of file to be uploaded e.g. "Pmts.txt", "CCTrans.zip"  | Yes       |
| DATA           | File data. Could be plain text, zip, or jpg. Base-64 encoded  | Yes       |
| START          | 0-based offset into file. This value should match the current size of the data received for the file so far. Defaults to 0 if not provided.                               | No        |

|              |   |     |
|--------------|---|-----|
| SIZE         | Size of data block (provided for error-checking purposes) and/or data-chunking purposes. Defaults to - 1 if not provided. | No  |
| FILESIZE     | Total size of file  | Yes |
| TEMPFILENAME | Name of temporary file (for subsequent data chunks)   |     |

### Output Parameters

| Parameter    | Description   |
|--------------|---|
| FILENAME     | The file's name as it exists on the server. This will be the original filename as provided in the Request FILENAME parameter with the date/time stamp and a possible 'tie-breaker' character appended (ie File20080904130423a.txt). This will only be sent when the file upload has been completed. |
| TEMPFILENAME | The path and name of the temporary file on the server. This will only be sent if the input parameters imply that there are additional data chunks to be sent.   |
| CURRENTSIZE  | Returns the current size of the file being uploaded.  |

### Usage Notes

The web service will store the file to be uploaded in the folder which serves as the source for the Flex handheld importer.

## T2\_FLEX\_IVR

### GetAllCitations() v1.0

This method retrieves information about multiple citations by license plate. Citations that have a \$0.00 amount due will also be returned by this web method.

### Input XML Parameters

| Parameter        | Description   | Required? |
|------------------|---|-----------|
| PLATE_NUMBER     | The plate number for the vehicle related to the citation    | Yes       |
| PLATE_STATE_UID  | The plate state UID for the vehicle related to the citation | Yes       |
| PLATE_TYPE_UID   | The plate type UID related to the vehicle plate             | No        |
| PLATE_SERIES_UID | The plate series UID related to the vehicle plate           | No        |

### Return XML Parameters

| Parameter | Description             |
|-----------|-------------------------|
| CON_UID   | The UID of the citation |

|                              |   |
|------------------------------|---|
| CON_SNAP_VEH_PLATE_LICENSE   | The snap vehicle plate license of the citation                |
| CON_SNAP_VEH_PLATE_REG_MONTH | The snap vehicle plate registration end month of the citation |
| CON_SNAP_VEH_PLATE_REG_YEAR  | The snap vehicle plate registration end year of the citation  |
| VIC_CODE                     | The violation code value of the citation                      |
| VIC_DESCRIPTION              | The violation description of the citation                     |
| CON_ISSUE_DATE               | The issue date of the citation                                |
| CON_AMOUNT_DUE               | The balance due of the citation                               |
| TAL_UID_TRANSFER_AGENCY      | The transfer agency UID of the citation                       |
| VPL_UID_SNAP_VEH_PLATE_TYPE  | The snap vehicle type UID of the citation                     |
| VPL_DESCRIPTION              | The vehicle plate type description of the citation            |
| STL_UID_SNAP_VEH_STATE       | The snap vehicle state UID of the citation                    |
| STL_DESCRIPTION              | The vehicle state description of the citation                 |
| CLM_UID_LOCATION             | The location UID of the citation                              |
| CLM_DESCRIPTION              | The location description of the citation                      |
| VIC_UID_VIOLATION_CODE       | The violation code UID of the citation                        |
| VIC_LONG_DESCRIPTION         | The violation code long description of the citation           |

### Usage Notes

If the Plate Number and Plate State do not provide a unique vehicle record, inputting the Plate Type and/or Plate Series is required.

## GetCitations() v1.0

This method retrieves information about a specific citation using the current SQL statement implemented by IVR.

### Input XML Parameters

| Parameter       | Description                          | Required? |
|-----------------|--------------------------------------|-----------|
| CITATION_NUMBER | Citation number issued to the parker | Yes       |

### Return XML Parameters

| Parameter                    | Description   |
|------------------------------|---|
| CON_UID                      | The UID of the citation                                       |
| CON_SNAP_VEH_PLATE_LICENSE   | The snap vehicle plate license of the citation                |
| CON_SNAP_VEH_PLATE_REG_MONTH | The snap vehicle plate registration end month of the citation |

|                             |  |
|-----------------------------|--|
| CON_SNAP_VEH_PLATE_REG_YEAR | The snap vehicle plate registration end year of the citation |
| VIC_CODE                    | The violation code value of the citation                     |
| VIC_DESCRIPTION             | The violation description of the citation                    |
| CON_ISSUE_DATE              | The issue date of the citation                               |
| CON_AMOUNT_DUE              | The balance due of the citation                              |
| TAL_UID_TRANSFER_AGENCY     | The transfer agency UID of the citation                      |
| VPL_UID_SNAP_VEH_PLATE_TYPE | The snap vehicle type UID of the citation                    |
| VPL_DESCRIPTION             | The vehicle plate type description of the citation           |
| STL_UID_SNAP_VEH_STATE      | The snap vehicle state UID of the citation                   |
| STL_DESCRIPTION             | The vehicle state description of the citation                |
| CLM_UID_LOCATION            | The location UID of the citation                             |
| CLM_DESCRIPTION             | The location description of the citation                     |
| VIC_UID_VIOLATION_CODE      | The violation code UID of the citation                       |
| VIC_LONG_DESCRIPTION        | The violation code long description of the citation          |

## T2\_FLEX\_LOYALTYPROGRAM

### EditLoyaltyCard

This Web method updates the editable details on a loyalty card.

#### Input XML Parameters

| Parameter Name          | Description   | Required? |
|-------------------------|---|-----------|
| LoyaltyCardUid          | The UID of the Loyalty Card being edited                          | Y         |
| PreferredFacilityUid    | The UID of the preferred facility set up for the loyalty program. | Y         |
| AutomaticallyUsePoints  | Y/N if points should be used automatically in the field.          | Y         |
| AutomaticallyChargeCard | Y/N if a Credit Card Profile should be used in the field.         | Y         |
| IsDeactivated           | Y/N if the Loyalty Card is deactivated.                           | Y         |
| CreditCardProfileUid    | The UID of a credit card profile to link to the loyalty card.     | N         |

**Returned XML Parameters**

| Parameter Name | Description   |
|----------------|---|
| Details        | The details of the edited loyalty card, contained in the <EditLoyaltyCardResult> outlined in the table below. |
| Errors         | A list of any errors encountered during execution.  |

**EditLoyaltyCardResult**

| Parameter Name           | Description  |
|--------------------------|--|
| LoyaltyCardUid           | UID of the loyalty card.   |
| LoyaltyCardNumber        | Loyalty card number.   |
| EntityUid                | Customer UID.  |
| EntityName               | Customer name.   |
| PrintOnline              | Y/N (should be set to No).   |
| SendByMail               | Y/N, whether the loyalty program for the card requires mailing.                  |
| IsInventoried            | Y/N, whether the card is inventoried. Should always be N.                        |
| RequiresMailing          | Y/N, whether the individual card requires mailing.                               |
| RequiresCardOnFile       | Y/N, whether card on file is required for this loyalty card.                     |
| PrintableCardTemplateUid | UID of the temporary printable card template.                                    |
| PreferredFacilityUid     | UID of the preferred facility.   |
| PreferredFacilityName    | Name of the preferred facility.  |
| TotalAccumulatedPoints   | Total number of points on the card that have ever been accumulated and redeemed. |
| RedeemablePoints         | Total number of points on the card that have not been redeemed.                  |
| RedeemableValue          | Value of the redeemable points.  |
| CurrentLoyaltyTierUid    | UID of current loyalty tier.   |
| CurrentLoyaltyTierName   | Name of current loyalty tier.  |

| Parameter Name          | Description   |
|-------------------------|---|
| TierMultiplier          | Multiplier for points earned while in the current tier.                         |
| AutomaticallyUsePoints  | Y/N whether redeemable value is automatically redeemed on the next transaction. |
| AutomaticallyChargeCard | Y/N whether the card on profile should be automatically charged.                |
| CreditCardProfileUid    | UID of the associated credit card profile.                                      |
| IsActive                | Y/N whether the loyalty card is active.   |

## GetActiveLoyaltyProgramDetails

This web method returns a list of details for a given active loyalty program in Flex.

### Input XML Parameters

| Parameter Name    | Description                                   | Required? |
|-------------------|---|-----------|
| LoyaltyProgramUid | The UID of an active loyalty program in Flex. | Y         |

### Returned XML Parameters

| Parameter Name | Description   |
|----------------|---|
| Details        | The details of the loyalty program, contained in the <GetActiveLoyaltyProgramDetailsOutputParameter> outlined in the table below. |
| Errors         | A list of any errors encountered during execution.  |

### GetActiveLoyaltyProgramDetailsOutputParameter

| Parameter Name        | Description   |
|-----------------------|---|
| Name                  | The Name of the loyalty program.  |
| PrintOnline           | Y/N if the loyalty card can be printed online.                            |
| SendByMail            | Y/N if the loyalty card can be sent by Mail.                              |
| AllocatedCashDrawers  | List of cash drawer UIDs that inventoried loyalty cards are allocated to. |
| RequiresCardOnFile    | Y/N if the customer must have a valid credit card profile.                |
| PrintableCardTemplate | UID of the template to use to print the loyalty card.                     |



| Parameter Name       | Description  |
|----------------------|--|
| IsGenerallyAvailable | Y/N if the loyalty card is generally available on the eBiz site or if it is considered a private (corporate) card. |

## GetActiveLoyaltyProgramDetailsAsXML

This web method returns a list of details for a given active loyalty program in Flex in the form of an XML string.

### Input XML Parameters

| Parameter Name    | Description                                   | Required? |
|-------------------|---|-----------|
| LoyaltyProgramUid | The UID of an active loyalty program in Flex. | Y         |

### Returned XML Parameters

| Parameter Name | Description  |
|----------------|--|
| Details        | The details of the loyalty program in an XML string. |
| Errors         | A list of any errors encountered during execution.   |

## GetFacilityPricingEstimate

This web method returns pricing for a given facility for a specified date/time.

### Input XML Parameters

| Parameter Name         | Description                           | Required? |
|------------------------|---------------------------------------|-----------|
| FacilityUid            | The UID of an active facility in Flex | Y         |
| EstimatedEntryDateTime | Estimated date/time of entrance.      | Y         |
| EstimateExitDateTime   | Estimated date/time of exit.          | Y         |

### Returned XML Parameters

| Parameter Name | Description   |
|----------------|---|
| Details        | The estimated parking cost contained in the <GetFacilityPricingEstimateResult> parameter. |
| Errors         | A list of any errors encountered during execution.  |

## GetLoyaltyCardAllocations

Retrieves a list of loyalty card allocations for a given cash drawer (or all web service cash drawers) and a specific loyalty program.

**Input XML Parameters**

| Parameter Name             | Description  | Required? |
|----------------------------|--|-----------|
| LoyaltyProgramUid          | The UID of the loyalty program to query.                       | Y         |
| CashDrawerLookupUid        | The UID of the cash drawer lookup to use to query allocations. | N         |
| SearchAllWebServiceDrawers | Y/N to search all web services cash drawers; default is false. | Y         |

**Returned XML Parameters**

| Parameter Name | Description  |
|----------------|--|
| Details        | The details of the requested loyalty card, contained in the <GetLoyaltyCardAllocationOutputParameter> outlined in the table below. |
| Errors         | A list of any errors encountered during execution.   |

**GetLoyaltyCardAllocationOutputParameter**

| Parameter Name     | Description                                   |
|--------------------|---|
| AllocationUid      | The UID of the allocation.                    |
| LoyaltyProgramUid  | The UID of the loyalty program.               |
| StartingAllocation | Starting numeric value in loyalty allocation. |
| EndingAllocation   | Ending numeric value in loyalty allocation.   |
| CashDrawerUid      | Cash drawer UID of allocation.                |

**GetLoyaltyCardDetails**

This Web method returns a list of loyalty card details for a specified customer and loyalty card.

**Input XML Parameters**

| Parameter Name | Description  | Required? |
|----------------|--|-----------|
| CustomerUid    | The UID of a customer. Either the customer UID or the loyalty card UID must be provided. If a customer UID is entered and a customer has multiple loyalty cards, all of them will be returned. | Y/N       |
| LoyaltyCardUid | The UID of a loyalty card associated with that customer. Either the customer UID or the loyalty card UID must be provided.   | Y/N       |

**Returned XML Parameters**

| Parameter Name | Description   |
|----------------|---|
| Details        | The details of the loyalty card, contained in the <LoyaltyCardOutputParameter> outlined in the table below. |
| Errors         | A list of any errors encountered during execution.  |

**LoyaltyCardOutputParameter**

| Parameter Name           | Description  |
|--------------------------|--|
| LoyaltyCardUid           | UID of the loyalty card.   |
| LoyaltyCardNumber        | Loyalty card number.   |
| EntityUid                | Customer UID.  |
| EntityName               | Customer name.   |
| PrintOnline              | Y/N (should be set to No).   |
| SendByMail               | Y/N, whether the loyalty program for the card requires mailing.                  |
| IsInventoried            | Y/N, whether the card is inventoried. Should always be N.                        |
| RequiresMailing          | Y/N, whether the individual card requires mailing.                               |
| RequiresCardOnFile       | Y/N, whether card on file is required for this loyalty card.                     |
| PrintableCardTemplateUid | UID of the temporary printable card template.                                    |
| PreferredFacilityUid     | UID of the preferred facility.   |
| PreferredFacilityName    | Name of the preferred facility.  |
| TotalAccumulatedPoints   | Total number of points on the card that have ever been accumulated and redeemed. |
| RedeemablePoints         | Total number of points on the card that have not been redeemed.                  |
| RedeemableValue          | Value of the redeemable points.  |
| CurrentLoyaltyTierUid    | UID of current loyalty tier.   |
| CurrentLoyaltyTierName   | Name of current loyalty tier.  |

| Parameter Name          | Description   |
|-------------------------|---|
| TierMultiplier          | Multiplier for points earned while in the current tier.                         |
| AutomaticallyUsePoints  | Y/N whether redeemable value is automatically redeemed on the next transaction. |
| AutomaticallyChargeCard | Y/N whether the card on profile should be automatically charged.                |
| CreditCardProfileUid    | UID of the associated credit card profile.                                      |
| IsActive                | Y/N whether the loyalty card is active.   |

### LinkUnlinkValuePermissionReservationToLoyaltyCard

This web method links or unlinks reservation value permits and loyalty cards.

#### Input XML Parameters

| Parameter Name     | Description  | Required? |
|--------------------|--|-----------|
| ValuePermissionUid | The UID of the reservation value permit to link/unlink.                        | Y         |
| LoyaltyCardUid     | The UID of the loyalty card to link/unlink.                                    | Y         |
| LinkCards          | 1 if the card and value permit should be linked, 0 if they should be unlinked. | Y         |

#### Returned XML Parameters

| Parameter Name | Description   |
|----------------|---|
| Details        | The details of the linked/unlinked card and value permit, contained in the <LinkUnlinkValuePermissionReservationToLoyaltyCardResult> outlined in the table below. |
| Errors         | A list of any errors encountered during execution.  |

#### LinkUnlinkValuePermissionReservationToLoyaltyCardResult

| Parameter Name     | Description  |
|--------------------|--|
| LoyaltyCardUid     | The UID of the linked/unlinked loyalty card.                         |
| ValuePermissionUid | The UID of the linked/unlinked value permit.                         |
| RelationshipUid    | The UID of the relationship between the card and value permit.       |
| LinkedCards        | 1 if the card and value permit were linked, 0 if they were unlinked. |

## RegisterLoyaltyCard

Registers an existing loyalty card for use with the system.

### Input XML Parameters

| Parameter Name       | Description   | Required? |
|----------------------|---|-----------|
| LoyaltyCardUid       | The UID of the loyalty card to register.                              | Y         |
| EntityUid            | The UID of the entity registering the card.                           | Y         |
| CreditCardProfileUid | The UID of the credit card profile associated with the card.          | N         |
| CashDrawerLookupUid  | The UID of the cash drawer lookup to use in the registration process. | Y         |

### Returned XML Parameters

| Parameter Name | Description  |
|----------------|--|
| Details        | The details of the requested loyalty card, contained in the <RegisterLoyaltyCardResult> outlined in the table below. |
| Errors         | A list of any errors encountered during execution.   |

### RegisterLoyaltyCardResult

| Parameter Name           | Description   |
|--------------------------|---|
| LoyaltyCardUid           | UID of the loyalty card.  |
| LoyaltyCardNumber        | Loyalty card number.  |
| EntityUid                | Customer UID.   |
| EntityName               | Customer name.  |
| PrintOnline              | Y/N (should be set to No).                                      |
| SendByMail               | Y/N, whether the loyalty program for the card requires mailing. |
| IsInventoried            | Y/N, whether the card is inventoried. Should always be N.       |
| RequiresMailing          | Y/N, whether the individual card requires mailing.              |
| RequiresCardOnFile       | Y/N, whether card on file is required for this loyalty card.    |
| PrintableCardTemplateUid | UID of the temporary printable card template.                   |

| Parameter Name          | Description  |
|-------------------------|--|
| PreferredFacilityUid    | UID of the preferred facility.   |
| PreferredFacilityName   | Name of the preferred facility.  |
| TotalAccumulatedPoints  | Total number of points on the card that have ever been accumulated and redeemed. |
| RedeemablePoints        | Total number of points on the card that have not been redeemed.                  |
| RedeemableValue         | Value of the redeemable points.  |
| CurrentLoyaltyTierUid   | UID of current loyalty tier.   |
| CurrentLoyaltyTierName  | Name of current loyalty tier.  |
| TierMultiplier          | Multiplier for points earned while in the current tier.                          |
| AutomaticallyUsePoints  | Y/N whether redeemable value is automatically redeemed on the next transaction.  |
| AutomaticallyChargeCard | Y/N whether the card on profile should be automatically charged.                 |
| CreditCardProfileUid    | UID of the associated credit card profile.                                       |
| IsActive                | Y/N whether the loyalty card is active.  |

## RequestLoyaltyCard

This Web method generates a new loyalty card or assigns a loyalty card by number to an entity.

### Input XML Parameters

| Parameter Name           | Description   | Required? |
|--------------------------|---|-----------|
| LoyaltyCardProgramUid    | UID of the loyalty program used to generate or assign a loyalty card.   | Y         |
| LoyaltyCardNumber        | The numeric loyalty card number to assign. The next available card number is generated if not provided.                                     | N         |
| LoyaltyCardAllocationUid | UID of the loyalty card allocation to use to generate the loyalty card. If not provided, the first valid allocation available will be used. | N         |
| EntityUid                | UID of the customer requesting or being   | Y         |

| Parameter Name      | Description  | Required? |
|---------------------|--|-----------|
|                     | assigned the card.   |           |
| FacilityUid         | UID of the facility that is the entity's preferred location for parking.   | Y         |
| CashDrawerLookupUid | UID of the cash drawer lookup value to use when generating or assigning a loyalty card. This value is used to determine the loyalty card allocation. If not provided, the method will search all web drawers for a valid allocation. | N         |
| SendByMail          | Y/N, whether to send the loyalty card by mail.   | Y         |

**Returned XML Parameters**

| Parameter Name | Description   |
|----------------|---|
| Details        | The details of the requested loyalty card, contained in the <RequestLoyaltyCardResult> outlined in the table below. |
| Errors         | A list of any errors encountered during execution.  |

**RequestLoyaltyCardResult**

| Parameter Name           | Description   |
|--------------------------|---|
| LoyaltyCardUid           | UID of the loyalty card.  |
| LoyaltyCardNumber        | Loyalty card number.  |
| EntityUid                | Customer UID.   |
| EntityName               | Customer name.  |
| PrintOnline              | Y/N (should be set to No).                                      |
| SendByMail               | Y/N, whether the loyalty program for the card requires mailing. |
| IsInventoried            | Y/N, whether the card is inventoried. Should always be N.       |
| RequiresMailing          | Y/N, whether the individual card requires mailing.              |
| RequiresCardOnFile       | Y/N, whether card on file is required for this loyalty card.    |
| PrintableCardTemplateUid | UID of the temporary printable card template.                   |

| Parameter Name          | Description  |
|-------------------------|--|
| PreferredFacilityUid    | UID of the preferred facility.   |
| PreferredFacilityName   | Name of the preferred facility.  |
| TotalAccumulatedPoints  | Total number of points on the card that have ever been accumulated and redeemed. |
| RedeemablePoints        | Total number of points on the card that have not been redeemed.                  |
| RedeemableValue         | Value of the redeemable points.  |
| CurrentLoyaltyTierUid   | UID of current loyalty tier.   |
| CurrentLoyaltyTierName  | Name of current loyalty tier.  |
| TierMultiplier          | Multiplier for points earned while in the current tier.                          |
| AutomaticallyUsePoints  | Y/N whether redeemable value is automatically redeemed on the next transaction.  |
| AutomaticallyChargeCard | Y/N whether the card on profile should be automatically charged.                 |
| CreditCardProfileUid    | UID of the associated credit card profile.                                       |
| IsActive                | Y/N whether the loyalty card is active.  |

### Usage Notes

If no loyalty card allocation specified, determine based on cash drawer passed in (or all web drawers if no cash drawer).

## TransferLoyaltyPoints

This Web method transfers a specified number of points from one specified loyalty card to another card belonging to the same customer.

### Input XML Parameters

| Parameter Name       | Description                                       | Required? |
|----------------------|---|-----------|
| LoyaltyCardSourceUid | The Loyalty Card UID to transfer the points from. | Y         |
| LoyaltyCardTargetUid | The Loyalty Card UID to transfer the points to.   | Y         |
| PointsToTransfer     | The number of point to transfer.                  | Y         |



**Returned XML Parameters**

| Parameter Name              | Description   |
|-----------------------------|---|
| TransferLoyaltyPointsResult | IsSuccessful: indicates with a 1 or 0 in the whether or not the points were successfully transferred.<br><br>PointsTransferred: returns the number of points transferred. |
| Errors                      | A list of any errors encountered during execution.  |

**Usage Notes**

Both the source and target loyalty cards must belong to the same customer.

**T2\_FLEX\_MAPS**

This Web service and its methods are for internal T2 Systems use only and should not be used by customers.

**T2\_FLEX\_MISC****CloseWebDrawer() v1.0**

This Web method closes a Web cash drawer at a specified time rather than after the next transaction after midnight comes in (standard web drawer close time).

**Input XML Parameters**

| Parameter     | Description                                       | Required |
|---------------|---|----------|
| CashDrawerUID | UID of the cash drawer to be closed.              | Yes      |
| CloseBalance  | Closing balance of the Web drawer.                | No       |
| CloseNote     | Note associated with the close of the Web drawer. | No       |

**Return XML Parameters**

| Parameter            | Description  |
|----------------------|--|
| CloseWebDrawerResult | A string indicating whether or not the web drawer was successfully closed. |

**ExecuteQuery() v1.0**

Note: See Using T2 Web Services to Query the T2 Flex Database in Chapter 2 for detailed instructions and examples.

**ExecuteQueryByName() v1.0**

Note: See Using T2 Web Services to Query the T2 Flex Database in Chapter 2 for detailed instructions and examples.

## InsertUpdateNotes() v1.0

This method is used to insert a Note associated with a customer, vehicle, permit, citation, or appeal record. Files such as photos may then be attached to the note using the InsertDocument() Web method (under T2\_Flex\_Reporting).

### Input XML Parameters

| Parameter            | Description  | Required                             |
|----------------------|--|--------------------------------------|
| NOTE_UID             | Note UID   | No, if inserting<br>Yes, if updating |
| NOTE_TYPE_UID        | Note Type Code UID.  | Yes, if inserting<br>No, if updating |
| NOTE_TEXT            | Note Text  | Yes, if inserting<br>No, if updating |
| NOTE_END_DATE        | Note end date. Format: MM/DD/YY  | No                                   |
| OBJECT_TYPE          | Object Type. Integer with possible values of<br>1 (for Entities), 3 (Vehicles), 10 (Permissions) 13 (Contraventions), or 14 (Adjudications). | Yes, if inserting<br>No, if updating |
| OBJECT_REFERENCE_UID | UID of the object to which the Note is attached  | Yes if inserting                     |

### Return XML Parameters

| Parameter | Description   |
|-----------|---|
| NOTE_UID  | UID of Note that was successfully inserted or updated |

### Usage Notes

If a Note UID is not passed in then the method will attempt to insert a new Note using the information passed into the method. If a Note UID is passed in then the method will attempt to update an existing note using information passed into the method.

## IsFlexAvailable() v1.0

### Input XML Parameters

None.

**Return XML Parameters**

| Parameter    | Description                           |
|--------------|---------------------------------------|
| IS_AVAILABLE | “1” if method was successfully called |

**Usage Notes**

This method is used for diagnostic purposes, to determine whether the T2 Flex Web Services can be called.

**UpdatePaymentPlanCustomFlds() v1.0****Input XML Parameters**

| Parameter Name | Description  | Required? |
|----------------|--|-----------|
| ENTITY_UID     | Customer UID of the customer associated with the payment plan. | Yes       |
| PAYPLAN_UID    | The Payment Plan UID to edit.                                  | Yes       |
| (FIELD_NAME)   | Custom Field Name (Ex: Cust_1)                                 | Yes       |

**Return XML Parameters**

| Parameter Name | Description   |
|----------------|---|
| PAYPLAN_UID    | The payment plan UID that was successfully updated. |

**Usage Notes**

This method allows the custom fields related to a Transfer Agency on an External payment plan to be edited after the payment plan has been created and before any payments have been transferred.

**T2\_FLEX\_MISCSALEITEMS****PayMiscSaleItem() v1.0**

This method allows you to execute a T2 Flex query using Web Services.

**Input XML Parameters**

| Parameter                  | Description  | Required                      |
|----------------------------|--|-------------------------------|
| MiscellaneousItems         | Contains a MiscellaneousItemListParam complex type* parameter set for each Misc item to be paid (see next item).   | Yes (one for each misc. item) |
| MiscellaneousItemListParam | Complex type* collection of the following parameters used to indicate which Misc Sale Item(s) are being paid: <ul style="list-style-type: none"> <li>MiscellaneousItemUID</li> </ul> | Yes                           |

|                                  |  |                                  |
|----------------------------------|--|----------------------------------|
| EntityUID                        | The customer UID paying for the item(s).   | No                               |
| PaymentMethodLookupUID           | The payment method UID.  | Yes                              |
| AmountPaid                       | The amount paid for the misc. item (allows two decimals).  | Yes                              |
| CashDrawerUID                    | The web service cash drawer UID for the payment.   | Yes                              |
| AccountCreditList                | Contains a AccountCreditsListParam complex type* parameter set for each credit on the customer's account (see next item).  | No (one for each account credit) |
| AccountCreditsListParam          | Complex type* collection of the following parameters used to indicate what credit(s) are on the customer's account: <ul style="list-style-type: none"> <li>AccountCreditsUID (integer)</li> </ul>              | No                               |
| ReferenceNote                    | A reference note for the payment.  | No                               |
| TransferAgencyLookupUID          | The agency UID to which payment for the sale item(s) is being transferred.   | No                               |
| PaymentScheduleTemplateLookupUID | The payment schedule template UID.   | No                               |
| PayplanStartDate                 | The date and time the payment plan starts in the format of YYYY-MM-DD.   | No                               |
| PayplanTimePeriodUid             | The payment plan time period UID.  | No                               |
| PayPlanTaxStatusLookupUID        | The payment plan tax status lookup UID.  | No                               |
| PayplanNumPayments               | The number of payments on the payment plan.<br><i>Note: If the amount per payment is specified in the XML, you cannot include the number of payments.</i>  | No                               |
| PayplanAmtPerPayment             | The amount per payment (allows two decimals).<br><i>Note: If the number of payments is specified in the XML, you cannot include the amount per payment.</i>  | No                               |
| PayplanCustomFields              | Contains a CustomFields complex type* parameter set for each custom field on the misc. item (see next item).   | No                               |
| CustomFields                     | Complex type* collection of the following parameters used to indicate any custom fields on the misc. item: <ul style="list-style-type: none"> <li>CustomName (string)</li> <li>CustomValue (string)</li> </ul> | No                               |

**Return XML Parameters**

| Parameter              | Description  |
|------------------------|--|
| PayMiscSaleItemsResult | Indicates in the results string whether or not the item was sold by including the MISCSALEITEM_LIST (with respective UIDs) and the RECEIPTUID. |

**Usage Notes**

\*Complex Type – This type of parameter contains nested parameters, as indicated in the table.

**T2\_FLEX\_OCCUPANCY****GetFacilityList() v1.0**

This Web method returns a list of known active facilities.

**Input XML Parameters**

Aside from standard T2 Web Services input parameters, this Web method does not have input parameters since it returns ALL known active facilities.

**Returned XML Parameters**

| Parameter Name        | Description   |
|-----------------------|---|
| GetFacilityListResult | Includes a facility UID and Description for each facility returned in the results string. |

**GetOccupancyData() v1.0**

This web method retrieves occupancy information for a selected facility or all facilities. Custom field data can be optionally returned as well.

**Input XML Parameters**

| Parameter Name  | Description   | Required? |
|-----------------|---|-----------|
| facilityUid     | The Facility UID for which you are retrieving occupancy data. To return results for every facility in the system, include a Facility UID of 0.                                    | No        |
| customFieldList | Complex type* collection of the following parameters used to indicate any custom field(s) on the facility: <ul style="list-style-type: none"> <li>String (Field Value)</li> </ul> | No        |

**Returned XML Parameters**

| Parameter Name         | Description   |
|------------------------|---|
| GetOccupancyDataResult | The results string includes a facility UID and Description for each facility returned as well as a data |

|  |   |
|--|---|
|  | set for each occupancy type that includes the following: <ul style="list-style-type: none"> <li>• Capacity (total number of spaces)</li> <li>• Occupied (number of filled spaces)</li> <li>• Available (number of available spaces)</li> <li>• Timestamp (date/time for the occupancy snapshot)</li> </ul> Information for the Parking Operation is also returned, including the site name and T2 Flex version. |
|--|---|

### Usage Notes

\*Complex Type – This type of parameter contains nested parameters, as indicated in the table.

## T2\_FLEX\_PERMISSIONS

### The VALID\_LOCATIONS Parameter

Some of the methods of the T2\_Flex\_Permissions Web service use the VALID\_LOCATIONS parameter. This parameter specifies one or more locations (unlimited) in which a permit is valid.

Note: This parameter is applied to every permit in the PERMISSION\_UID\_LIST parameter, and each location must be valid for every permit.

This parameter is an XML string, formatted as shown in the following example:

```
<Parameter>
<Name>VALID_LOCATIONS_LIST</Name>
<Value>
  <ValidLocations>
    <Location>
      <facility_UID>2000</facility_UID>
      <area_UID>2010</area_UID>
      <stallType_UID>2005</stallType_UID>
      <stall_UID></stall_UID>
    </Location>
    <Location>
      <facility_UID>2001</facility_UID>
      <area_UID>2011</area_UID>
      <stallType_UID>2005</stallType_UID>
      <stall_UID>2007</stall_UID>
    </Location>
  </ValidLocations>
</Value>
</Parameter>
```

The fields comprising a Location are as follows:

| Parameter    | Description                     | Required |
|--------------|---------------------------------|----------|
| FACILITY_UID | Facility UID (FACILITY.FAC_UID) | Yes      |

|               |   |  |
|---------------|---|--|
| AREA_UID      | Area UID (AREA.ARE_UID)                         | Yes, if a stall type UID is passed. Otherwise, no. |
| STALLTYPE_UID | Stall type UID (PARKING_STALL_TYPE_LKP.PST_UID) | Yes, if a stall UID is passed. Otherwise, no.      |
| STALL_UID     | Stall UID (PARKING_STALL.PKS_UID)               | No   |
| TIER_UID      | Facility Tier UID (FACILITY_TIER.FAL_UID)       | Yes, if the control group uses facility tiers.     |

## AddPermissionFee() v1.0

### Input XML Parameters

| Parameter          | Description                        | Required |
|--------------------|------------------------------------|----------|
| PERMISSION_UID     | Permit UID (PERMISSION.PER_UID)    | Yes      |
| PERMISSION_FEE_UID | Permit Fee UID (MISC_ITEM.MIS_UID) | Yes      |

### Return XML Parameters

| Parameter          | Description                         |
|--------------------|-------------------------------------|
| PERMISSION_UID     | Permit UID that was passed as input |
| PERMISSION_BALANCE | Updated permit balance              |

### Usage Notes

This method is used to add a fee to a permit at the same time as the Apply Fee menu option in T2 Flex is used. It also allows an unrenewed permit fee to be added to a permit that has not been renewed before its grace period has expired. It uses the fee schedule of type value credential for a value credential permit passed in to determine the permits price.

The MISC\_ITEM.MIS\_UID (Permit Fee UID) must be of type “Permit”, which is designated as having an MFC\_UID\_MISC\_FEE\_CATEGORY value of 4.

## AddPermissionToProfileRelationship() v1.0

### Input XML Parameters

| Parameter Name | Description                            | Required? |
|----------------|--|-----------|
| PERMIT_UID     | The permission the Credit Card Profile | Yes       |

|             |  |     |
|-------------|--|-----|
|             | will be linked to.   |     |
| PROFILE_UID | The Credit Card Profile to be linked to the permission.  | Yes |
| STARTDATE   | The starting date for the relationship.<br>Note: when this is not provided the StartDate will be stored as the current date. | No  |
| ENDDATE     | The ending date for the relationship.  | No  |

**Returned XML Parameters**

| Parameter Name                      | Description  |
|-------------------------------------|--|
| PERMISSION_PROFILE_RELATIONSHIP_UID | The Permission to Credit Card Profile Relationship that was added. |

**Usage Notes**

This web method will be called to link a credit card profile to an auto-renewable permit in the T2 Flex database.

**AddProfile() v1.0****Input XML Parameters**

| Parameter Name              | Description   | Required? |
|-----------------------------|---|-----------|
| ENTITY_UID                  | The customer the profile should be created for.   | Yes       |
| CREDIT_PAYMENT_METHOD       | The Credit Card Payment Method as defined in T2 Flex.   | Yes       |
| CREDIT_CARD_EXPIRATION_DATE | The date the Credit Card Expires.<br>Note: Stored date will default to the first of the month.      | Yes       |
| PROFILE_TOKEN               | The IPG Generated Token for the Profile.<br>Note: T2 Flex currently only supports Chase Paymentech. | Yes       |
| FIRSTFOUROFCARD             | The first four digits on the credit card.   | No        |
| LASTFOUROFCARD              | The last four digits on the credit card.  | Yes       |
| FIRSTNAMEONCARD             | The first name on the credit card.  | No        |
| LASTNAMEONCARD              | The last name on the credit card.   | No        |



### Returned XML Parameters

| Parameter Name | Description  |
|----------------|--|
| PROFILE_UID    | The unique identifier of the Credit Card Profile that was added. |

### Usage Notes

This web method will be called to add a new Credit Card Profile to the T2 Flex database. The supplied payment method will be checked to ensure it supports credit card profiles.

## AddUnitsToValueCredential() v1.0

### Input XML Parameters

| Parameter        | Description  | Required?             |
|------------------|--|-----------------------|
| Permission_UID   | The UID of the value credential to add value to                              | Yes                   |
| Number_Of_Units  | Number of days, uses, or dollar amount of value to place on value credential | Yes                   |
| Payment_Type_UID | The UID of the payment type in Flex to use to pay for the value              | Yes                   |
| Entity_UID       | The UID of the customer in Flex tied to the value credential                 | Yes                   |
| UserName         | Standard WS Username   | Yes                   |
| Password         | Standard WS Password   | Yes                   |
| Reference_Notes  | Any special notes related to the transaction                                 | Yes, but can be blank |
| Cash_Drawer_UID  | The cash drawer to use to sell the value credential                          | Yes                   |

### Return XML Parameters

| Parameter   | Description   |
|-------------|---|
| NewValue    | The value of the card after the number of units has been added. |
| Receipt_UID | The receipt UID from the sale of the number of units added.     |

### Usage Notes

This method emulates the **Add Value** menu option in the permit manager. It sells Value Credential Value and adds value to value credential. The value credential must have a value credential fee schedule.

**AuthPermissionSale() v1.1****Input XML Parameters**

| Parameter         | Description  | Required  |
|-------------------|--|---|
| PERMISSION_UID    | UID (PERMISSION.PER_UID) of permit to be sold. The permit must be reserved first.  | Yes   |
| ENTITY_UID        | UID (ENTITY.ENT_UID) of customer to which the permit is to be sold.  | Yes   |
| CONTROL_GROUP_UID | Control Group UID (PERMISSION_CONTROL_GROUP.PEC_UID).  | Yes   |
| VALID_LOCATIONS   | XML formatted string containing facilities, areas, stall types, and stalls where a permit is valid (see further explanation at the beginning of this section). | No, unless the control group uses facility tiers. This parameter will be ignored for pre-sold events. |

**Return XML Parameters**

| Parameter             | Description  |
|-----------------------|--|
| PERMISSION_UID        | UID of permit that is authorized for sale to the entity. Will be empty if the control group is part of a non-inventoried number range. See Usage Notes below for more information on non-inventoried ranges. |
| CONTRAVENTION_BALANCE | Citation balance due for this entity.  |
| CONTRAVENTION_COUNT   | Number of citations attached to this customer that are unpaid; that have not been transferred or voided; and that are not currently under appeal.  |

**Usage Notes**

This Web Method validates that the purchasing customer is eligible to purchase that permit. The method checks the following criteria for inventoried permits:

- The permit has been reserved for the customer passed into the method.
- The permit is able to be sold.
- The permit control group is available for Web sales.\*
- The customer is of the correct subclassification for the control group.\*

- The customer has not surpassed the number of permits available for sale to him or her.
- The permit sale does not violate any sales capacity limits.
- The sale does not violate any Waiting List rules.
- The permit district is not restricted by the control group.\*

**Note:** If the “Residential Permit Sales Based On Property” configuration setting is enabled, identify the property to assign to the permit. This property should have a permit district that meets the criteria defined by the control group.

- The permit sale does not violate Permit Allotment rules.
- The supplied Valid Locations parameter meets the appropriate business rules.\*

\*Only these criteria are checked for non-inventoried permits. In addition, all business rules relating to the permit UID will be ignored.

The method also reports the number and value of any citations assigned to a customer. The program calling this method must determine whether the number and balance of citations should prevent the permit sale.

Your organization may have additional permit eligibility business rules other than those verified by this method. Checking for those additional criteria is the responsibility of the calling program. When this method is called, it will check to determine if any external Eligibility Rules (determined in Permit System Settings) apply to the permit being authorized.

**Note:** If your organization’s business rules conflict with the criteria listed here, do not call this method, and instead implement eligibility checks by querying the T2 Flex database directly from your program.

- If the Eligibility Rule is **External Rules Only**, the method calls the external Single Permit Eligibility Web Method to determine eligibility. If the permit is not eligible, the error returned will be the text value of the NOT\_ELIGIBLE\_REASON parameter from the external method.
- If the Eligibility Rule is **T2 Flex Rules External Override**, then T2 Flex rules are sent to the external Single Permit Eligibility Web Method to determine if those eligibility rules are met. If the permit is not eligible, the error returned will be based on the XML return parameters of the external method.

**Note:** See [Chapter 4](#) for more detail on requirements for the external permit eligibility Web Methods.

## AvailableNumberRangesForEntity() v1.0

This Web method returns a list of permit number ranges available for sale to a given customer.

### Input XML Parameters

| Parameter | Description                               | Required |
|-----------|---|----------|
| EntityUid | UID of the customer for which the list of | Yes      |

|  |                          |  |
|--|--------------------------|--|
|  | PNRs is being retrieved. |  |
|--|--------------------------|--|

**Return XML Parameters**

| Parameter                            | Description                                   |
|--------------------------------------|---|
| AvailableNumberRangesForEntityResult | A string listing the available number ranges. |

**ClearPermissionCustody() v1.0**

This web method method that allows the caller to manually clear custody from a permit. The web user must have the Permit > Clear Custody user privilege.

**Input XML Parameters**

| Parameter Name | Description                                      | Required? |
|----------------|--|-----------|
| PERMISSION_UID | The permit for which to clear the custody value. | Yes       |

**Returned XML Parameters**

| Parameter Name | Description   |
|----------------|---|
| PERMISSION_UID | The permit for which the custody value was cleared. |

**Processing Logic**

Permits must be under strict inventory control and have a status of Confirmed Inactive, Unconfirmed Inactive, or Active.

The custody type assigned to the permit will be the system default (0 – Default)

**ConfirmPermissionPossession() v1.0****Input XML Parameters**

| Parameter      | Description   | Required |
|----------------|---|----------|
| PERMISSION_UID | Permit UID (PERMISSION.PER_UID) of the permit to confirm.               | Yes      |
| RECEIPT_UID    | Receipt UID (RECEIPT.REC_UID) from the original purchase of the permit. | Yes      |

**Return XML Parameters**

| Parameter      | Description                                 |
|----------------|---|
| PERMISSION_UID | Permit UID that was successfully confirmed. |

**Usage Notes**

This method is used to update a permit requiring possession confirmation to indicate that confirmation was received.

The following process is used to confirm possession of a permission:

- The permit and receipt are validated for their existence.
- The method checks that the permit is on the receipt.
- The method checks that the permit requires confirmation and does not have a confirmation date on it.
- The method sets the confirmation date to the current date on the permission.

## ConfirmPermissionReplacement() v1.0

### Input XML Parameters

| Parameter        | Description                       | Required? |
|------------------|-----------------------------------|-----------|
| PERMISSION_UID   | Permit ID                         | Yes       |
| RENEW_PERMISSION | Renew Permit? 1 for Yes; 0 for No | Yes       |

### Return XML Parameters

| Parameter      | Description                                 |
|----------------|---|
| PERMISSION_UID | Permit UID that was successfully confirmed. |

### Usage Notes

This method is used to confirm whether or not users want to renew their permit when it is up for renewal to replace the existing permit.

## DeletePermissionToProfileRelationship() v1.0

### Input XML Parameters

| Parameter Name                      | Description   | Required? |
|-------------------------------------|---|-----------|
| PERMISSION_PROFILE_RELATIONSHIP_UID | The Permission to Credit Card Profile Relationship to be edited.  | Yes       |
| PERMIT_UID                          | The permission that the Permission to Credit Card Profile Relationship is associated with.<br>Note: this is used to ensure the Relationship being edited is for the permission specified. | Yes       |

### Returned XML Parameters

| Parameter Name | Description                                    |
|----------------|--|
| SUCCESS        | Whether the operation was successful. [1 or 0] |

### Usage Notes

This web method will be called to remove an existing Permit Credit Card Profile Relationship from the T2 Flex database.

## DeleteProfile() v1.0

### Input XML Parameters

| Parameter Name | Description   | Required? |
|----------------|---|-----------|
| ENTITY_UID     | The customer the profile should be created for.<br><br>Note: this is used to ensure the Profile being edited is for the entity specified. | Yes       |
| PROFILE_UID    | The Credit Card Profile to be deleted.  | Yes       |

### Returned XML Parameters

| Parameter Name | Description                                    |
|----------------|--|
| SUCCESS        | Whether the operation was successful. [1 or 0] |

### Usage Notes

This web method will be called to delete an existing Credit Card Profile in the T2 Flex database. Deleting a Credit Card Profile will also delete all the Permit Credit Card Profile Relationships that use the target Credit Card Profile.

## EditProfile

### Input XML Parameters

| Parameter Name              | Description   | Required? |
|-----------------------------|---|-----------|
| ENTITY_UID                  | The customer the profile should be created for.<br><br>Note: this is used to ensure the Profile being edited is for the entity specified. | Yes       |
| CREDIT_PAYMENT_METHOD       | The Credit Card Payment Method as defined in FLEX.  | Yes       |
| PROFILE_UID                 | The Credit Card Profile to be edited.   |           |
| CREDIT_CARD_EXPIRATION_DATE | The date the Credit Card Expires.<br>Note: Stored date will default to the first of the month.  | Yes       |
| PROFILE_TOKEN               | The IPG Generated Token for the Profile.<br>Note: FLEX currently only   | Yes       |

|                 |   |     |
|-----------------|---|-----|
|                 | supports Chase Paymentech.                |     |
| FIRSTFOUROFCARD | The first four digits on the credit card. | Yes |
| LASTFOUROFCARD  | The last four digits on the credit card.  | Yes |
| FIRSTNAMEONCARD | The first name on the credit card.        | Yes |
| LASTNAMEONCARD  | The last name on the credit card.         | Yes |

**Returned XML Parameters**

| Parameter Name | Description   |
|----------------|---|
| PROFILE_UID    | The unique identifier of the Credit Card Profile that was edited. |

**Usage Notes**

This web method will be called to edit an existing Credit Card Profile in the T2 Flex database. The supplied credit card profile is validated to be associated with the supplied customer, and payment method is checked to ensure it supports credit card profiles.

**GetCurrentValueOfValueCredential() v1.0****Input XML Parameters**

| Parameter      | Description                     | Required? |
|----------------|---------------------------------|-----------|
| PERMISSION_UID | The UID of the value credential | Yes       |
| USERNAME       | Standard WS Username            | No        |
| PASSWORD       | Standard WS Password            | No        |

**Return XML Parameters**

| Parameter    | Description                                 |
|--------------|---|
| CurrentValue | The remaining value of the value credential |

**Usage Notes**

This method retrieves the current remaining value on a value credential.

**GetPermissionAlternateIDs() v1.0****Input XML Parameters**

| Parameter      | Description | Required                                   |
|----------------|-------------|--|
| PERMISSION_UID | Permit UID  | Yes, unless PERMISSION_NUMBER is supplied. |

|                   |               |   |
|-------------------|---------------|---|
| PERMISSION_NUMBER | Permit Number | Yes, unless PERMISSION_UID is supplied. |
|-------------------|---------------|---|

**Return XML Parameters**

| Parameter                | Description                           |
|--------------------------|---------------------------------------|
| ALTERNATE_ID_DESCRIPTION | Alt ID field description from T2 Flex |
| ALTERNATE_ID_NUMBER      | Alt ID number                         |
| ALTERNATE_ID_ACTIVE      | True or false if alt ID is active     |

**Usage Notes**

The XML Parameters can be repeated for each Alt ID found.

**GetPermissionBalance() v1.0****Input XML Parameters**

| Parameter      | Description | Required |
|----------------|-------------|----------|
| PERMISSION_UID | Permit UID  | Yes      |

**Return XML Parameters**

| Parameter          | Description                         |
|--------------------|-------------------------------------|
| PERMISSION_UID     | PERMIT UID that was passed as input |
| PERMISSION_BALANCE | Balance remaining on permission     |

**Usage Notes**

This method returns the current balance for a permit. This method can only be used for permits with a workflow status of Active, Returned, or Unconfirmed Inactive.

**GetPermissionPrice() v1.1****Input XML Parameters**

| Parameter                 | Description  | Required   |
|---------------------------|--|--|
| PERMISSION_UID            | Permit UID (PERMISSION.PER_UID) that was passed as input | Yes  |
| CONTROL_GROUP_UID         | Control Group UID (PERMISSION_CONTROL_GROUP.PEC_UID)     | Yes, unless reserving an Event permit (then PPQ_UID_PRESOLD_PEC_REL is required) |
| PERMISSION_EFFECTIVE_DATE | Permit effective date. Format is MM/DD/YYYY              | No   |



|                            |   |                                   |
|----------------------------|---|-----------------------------------|
| PERMISSION_EXPIRATION_DATE | Permit expiration date. Format is MM/DD/YYYY                                  | No                                |
| PPQ_UID_PRESOLD_PEC_REL    | The Event Facility-Presold PCG relationship the permit will be purchased from | Yes, if reserving an Event permit |

### Return XML Parameters

| Parameter        | Description          |
|------------------|----------------------|
| PERMISSION_UID   | Permit UID           |
| PERMISSION_PRICE | Current permit price |

### Usage Notes

This method returns the current price for an unsold permit using the appropriate fee schedule type depending on the type of permit.

For permits with a “Set at Time of Sale” effective and/or expiration date, the permit effective date and/or permit expiration date must be passed in order to receive the correct permit price. If these dates are not entered the method will return the price of the permit for one day. For permits with a preset effective or expiration date these parameters will be ignored.

## GetPermissionPriceByPCG() v1.2

### Input XML Parameters

| Parameter                  | Description  | Required  |
|----------------------------|--|---|
| PERMISSION_EFFECTIVE_DATE  | Effective Date if needed for Control Group                                   | No  |
| PERMISSION_EXPIRATION_DATE | Expiration Date if needed for control group                                  | No  |
| ENTITY_UID                 | Customer UID   | Yes   |
| CASHDRAWER_UID             | Cashier Station UID  | Yes   |
| PERMISSION_CTL_GRP_LIST    | List of Control Group UIDs   | Yes   |
| NUMBER_OF_UNITS            | Number of days, uses, or dollar amount of value to place on value credential | No, unless control group.number.range.permit.type.physicalgroup.type is value credential and if value credential has value credential fee type schedule |

|                              |  |    |
|------------------------------|--|----|
| PPQ_UID_PRESOLD_PEC_REL_LIST | The Event Facility-Presold PCG relationship list used to retrieve prices | No |
|------------------------------|--|----|

**Return XML Parameters**

| Parameter | Description   |
|-----------|---|
| UID       | Price associated with Control Group UID, followed by the values for the applied taxes in a comma separated list like the following:<br>PRICE,TAX2,TAX2,TAX3 |

**Usage Notes**

This method uses the Number\_Of\_Units passed in for all types of permit control groups (inventoried and non-inventoried) and returns the appropriate price for each.

**GetUnitCostOfValueCredentialByPCG() v1.0****Input XML Parameters**

| Parameter         | Description  | Required? |
|-------------------|--|-----------|
| Control_Group_UID | The UID of the value credential control group to get the cost per unit from. | Yes       |
| UserName          | Standard Web Services username.  | Yes       |
| Password          | Standard Web Services password.  | Yes       |
| CASH_DRAWER_UID   | The UID of the cash drawer.  | Yes       |
| Entity_UID        | The UID of the customer.   | Yes       |

**Return XML Parameters**

| Parameter   | Description   |
|-------------|---|
| CostPerUnit | The value of the card after the number of units has been added. |

**Usage Notes**

This method emulates the add value context menu in the permit manager and returns the cost per unit for the value credential.

It uses the subclassification of the customer if the customer is passed in, otherwise it will return the default cost of the control group's fee schedule.

**InsertUpdateCustomFlds() v1.0****XML Parameters**

| Parameter      | Description                     | Required |
|----------------|---------------------------------|----------|
| (FIELD_NAME)   | Custom Field Name Ex: Cust_1    | Yes      |
| PERMISSION_UID | Permit UID (PERMISSION.PER_UID) | Yes      |

**Return XML Parameters**

| Parameter      | Description |
|----------------|-------------|
| PERMISSION_UID | Permit UID  |

**Usage Notes**

This method populates an existing custom field on the permit record specified by PERMISSION\_UID. The parameter name designated by {FIELD\_NAME} is the name of the custom field to be used. The parameter value is the value to be inserted.

More than one field can be passed by including multiple field names among the XML parameters.

**InsertUpdatePermissionRelationship() v1.0****Input XML Parameters**

| Parameter               | Description  | Required |
|-------------------------|--|----------|
| PERMISSION_UID          | Permit UID (PERMISSION.PER_UID)  | Yes      |
| ENTITY_UID              | Customer UID (ENTITY.ENT_UID) to link with permit. Pass this parameter only if inserting a new customer relationship for the permit.   | No       |
| VEHICLE_UID             | Vehicle UID (VEHICLE.VEH_UID) to link with permit. Pass this parameter only if inserting a new vehicle relationship for the permit.    | No       |
| PROPERTY_UID            | Property UID (PROPERTY.PRO_UID) to link with permit. Pass this parameter only if inserting a new property relationship for the permit. | No       |
| ENTITY_RELATIONSHIP_UID | Customer Relationship UID (PER_ENT_REL.PNR_UID). Pass this parameter only if updating the dates for an existing customer relationship. | No       |

|                           |  |     |
|---------------------------|--|-----|
| VEHICLE_RELATIONSHIP_UID  | Vehicle Relationship UID (PER_VEH_REL.PVR_UID). Pass this parameter only if updating the dates for an existing vehicle relationship.   | No  |
| PROPERTY_RELATIONSHIP_UID | Property Relationship UID (PER_PRO_REL.PPR_UID). Pass this parameter only if updating the dates for an existing property relationship. | No  |
| RELATIONSHIP_START_DATE   | Relationship start date. Format is MM/DD/YYYY.   | Yes |
| RELATIONSHIP_END_DATE     | Relationship end date. Format is MM/DD/YYYY.   | No  |

### Return XML Parameters

| Parameter                 | Description  |
|---------------------------|--|
| PERMISSION_UID            | Permit UID for which relationships were inserted or updated.       |
| ENTITY_RELATIONSHIP_UID   | UID of customer relationship that was inserted or updated, if any. |
| VEHICLE_RELATIONSHIP_UID  | UID of vehicle relationship that was inserted or updated, if any.  |
| PROPERTY_RELATIONSHIP_UID | UID of vehicle relationship that was inserted or updated, if any.  |

### Usage Notes

This method is used to insert or update relationships for a permit. It is only possible to add or modify relationships for a permit that has been sold (that is, a permit whose status is not Reserved, Inventory, or Allocated).

A single call to this method may be used to insert one or more relationships, or update one more relationships, but not both. If you need to insert a relationship for a permit, and also update an existing relationship, you must make two separate calls to this method.

The start and end dates passed into this method will be applied to all relationships being inserted or updated. If you need to insert or update two different relationships such that they have different start or end dates, you must make two separate calls to this method.

Note that at least one of ENTITY\_UID, VEHICLE\_UID, PROPERTY\_UID, ENTITY\_RELATIONSHIP\_UID, VEHICLE\_RELATIONSHIP\_UID, or PROPERTY\_RELATIONSHIP\_UID is required as an input parameter. If neither an object UID nor object relationship UID is passed, an error will be returned.

## OverridePermissionCustody() v1.0

This Web method allows the caller to manually assign custody to a permit. The web user must have the Permit > Override Custody user privilege.

### Input XML Parameters

| Parameter Name | Description  | Required? |
|----------------|--|-----------|
| PERMISSION_UID | The permit for which to override custody.              | Yes       |
| USER_UID       | The user to which to assign custody of the permission. | Yes       |

### Return XML Parameters

| Parameter Name | Description   |
|----------------|---|
| PERMISSION_UID | The permit for which the custody value was overridden.  |
| USER_UID       | The user for which the custody of the permit was given. |

### Usage Notes

Permits must be under strict inventory control and have a status of Confirmed Inactive, Unconfirmed Inactive, or Active.

The custody type assigned to the permit will be Held for Distribution.

## PayPermissionBalance() v1.1

### Input XML Parameters

| Parameter           | Description   | Required                                     |
|---------------------|---|--|
| PERMISSION_UID      | Permit UID (PERMISSION.PER_UID)   | Yes  |
| CASH_DRAWER_UID     | Cash Drawer UID<br>(CASHDRAWER_LKP.CDL_UID)                             | Yes  |
| ENTITY_UID          | Customer UID (ENTITY.ENT_UID)<br>responsible for balance                | Yes  |
| AMOUNT_PAID         | Amount paid by customer, including tax<br>Format is decimal, e.g., 6.25 | Yes  |
| PAYMENT_TYPE_UID    | Payment Method UID<br>(PAYMENT_METHOD_MLKP.PMM_UID)                     | Yes  |
| REFERENCE_NOTE      | Reference Note on payment   | No   |
| ACCOUNT_CREDIT_LIST | A comma separated list of Account Credits<br>(ACCOUNT_CREDITS.ACR_UID)  | Yes, if<br>account<br>payment<br>method used |
| (FIELD_NAME)        | Custom Field Name (Ex: Cust_1)  | No   |

**Return XML Parameters**

| Parameter      | Description                         |
|----------------|-------------------------------------|
| PERMISSION_UID | Permit UID that was passed as input |
| RECEIPT_UID    | UID of receipt for this payment     |

**Usage Notes**

This method is used to pay the balance of a permit of any status, inactive, issued, and expired. An error will be returned if there is no balance on the permit.

Note that this method is **not** to be used for the initial sale of a permit; for that purpose, use the SellPermission method instead.

When external payment plan is the payment method, custom fields associated with the transfer agency used on the payment plan are not required.

**RenewPermission() v1.1****Input XML Parameters**

| Parameter              | Description   | Required                               |
|------------------------|---|--|
| ENTITY_UID             | Customer UID of permit holder   | Yes                                    |
| PERMISSION_UID         | Permit UID (PERMISSION.PER_UID)   | Yes                                    |
| RENEWAL_PERIOD_UID     | Permit renewal schedule UID (PERMISSION_RENEWAL_SCHED.PRS_UID)            | Yes                                    |
| RENEWAL_EFFECTIVE_DATE | Effective renewal date. Format: MM/DD/YYYY                                | Yes                                    |
| CASH_DRAWER_UID        | Cash Drawer UID (CASHDRAWER_LKP.CDL_UID)                                  | Yes                                    |
| AMOUNT_PAID            | Amount paid by customer, including tax. Format is decimal, e.g., 6.25     | Yes                                    |
| PAYMENT_TYPE_UID       | Payment Method UID (PAYMENT_METHOD_MLKP.PMM_UID), including payment plans | Yes                                    |
| REFERENCE_NOTE         | Reference Note on payment   | No                                     |
| ACCOUNT_CREDIT_LIST    | A comma separated list of Account Credits (ACCOUNT_CREDITS.ACR_UID)       | Yes, if account payment method is used |

|                           |  |    |
|---------------------------|--|----|
| Custom Payment Plan Field | Custom payment plan field, if a custom payment plan should be used as the payment method | No |
|---------------------------|--|----|

### Return XML Parameters

| Parameter      | Description                         |
|----------------|-------------------------------------|
| PERMISSION_UID | UID of permit that was renewed      |
| RECEIPT_UID    | UID of receipt for this transaction |

### Usage Notes

This method will attempt to renew a permit. After validating the required parameters the permit will be renewed, added to the selection basket, and the sale completed.

On the successful completion of this method an XML string will be returned containing the permit UID, and the UID of the receipt.

## ReservePermission() v2.1

### Input XML Parameters

| Parameter                 | Description   | Required?                            |
|---------------------------|---|--------------------------------------|
| PERMISSION_UID            | Permit UID (PERMISSION.PER_UID) to reserve. If an UID is not included the next available permit will be reserved. | No, if Perm_Num_Range_UID is entered |
| PERMISSION_NUM_RANGE_UID  | Permit Number Range UID (PERMISSION_NUMBER_RANGE.PNA_UID).  | No, if Permission_UID is entered     |
| CONTROL_GROUP_UID         | Control Group UID (PERMISSION_CONTROL_GROUP.PEC_UID).   | Yes                                  |
| CASH_DRAWER_UID           | Cash Drawer UID (CASHDRAWER_LKP.CDL_UID) to reserve permit from.  | Yes                                  |
| ENTITY_UID                | Customer UID (ENTITY.ENT_UID)   | Yes                                  |
| START_DATE                | Start date of reservation. If no value is passed in the current date will be used. Format is MM/DD/YYYY.          | No                                   |
| END_DATE                  | End date of reservation. Format is MM/DD/YYYY.  | Yes                                  |
| PERMISSION_EFFECTIVE_DATE | Permit effective date. Format is MM/DD/YYYY. Used to calculate price of an event permit.                          | Yes, if reserving an Event permit    |
| PERMISSION_EXPIRATION     | Permit expiration date. Format is   | Yes, if reserving                    |

|        |   |                 |
|--------|---|-----------------|
| N_DATE | MM/DD/YYYY. Used to calculate price of an event permit. | an Event permit |
|--------|---|-----------------|

### Return XML Parameters

| Parameter                  | Description  |
|----------------------------|--|
| PERMISSION_UID             | UID of permit that was successfully reserved.                |
| PERMISSION_NUMBER          | Number of permit that was successfully reserved.             |
| PERMISSION_PRICE           | Price of permit that was successfully reserved.              |
| PERMISSSION_TAX_1          | Tax amount #1 due for permit that was successfully reserved. |
| PERMISSSION_TAX_2          | Tax amount #2 due for permit that was successfully reserved. |
| PERMISSSION_TAX_3          | Tax amount #3 due for permit that was successfully reserved. |
| PERMISSION_EFFECTIVE_DATE  | Effective date of permit that was successfully reserved.     |
| PERMISSION_EXPIRATION_DATE | Expiration date of permit that was successfully reserved.    |

### Usage Notes

This method is used to reserve a permit for subsequent sale. Sale of the permit is accomplished by calling the SellPermission method on the permit that was reserved.

If a permit UID is passed, then the method will attempt to reserve that specific permit. However, if a permit number range is passed, then the next available permit will be reserved. ("Next available" is determined by choosing the first in a list of permits, ordered by UID, that have been allocated to the specified cash drawer from the specified permit number range.)

For permits with a "Set at Time of Sale" effective and/or expiration date, the permit effective date and/or permit expiration date must be passed in order to receive the correct permit price. If these dates are not entered the method will return the price of the permit for one day. For permits with a preset effective or expiration date these parameters will be ignored.

When reserving strict inventory controlled permits that are allocated to a Web services cash drawer, the current Web user must have custody.



**ReservePermissionQuantity() v1.1***Input XML Parameters*

| Parameter                   | Description  | Required?   |
|-----------------------------|--|---|
| PERMISSION_NUM_RANGE_UID    | Permit Number Range UID (PERMISSION_NUMBER_RANGE.PNA_UID)  | Yes   |
| CONTROL_GROUP_UID           | Control Group UID (PERMISSION_CONTROL_GROUP.PEC_UID)   | Yes   |
| QUANTITY                    | The number of permits to reserve from the PERMISSION_NUM_RANGE_UID   | Yes   |
| CASH_DRAWER_UID             | Cash Drawer UID (CASHDRAWER_LKP.CDL_UID)   | Yes   |
| ENTITY_UID                  | Customer UID (ENTITY.ENT_UID)  | No  |
| RELATIVE_RESERVATION_LENGTH | The number of minutes for which the reservation should be valid. Minimum length is 5 minutes, and it should at least be longer than the time it takes to complete the sale. This value determines the END_DATE_TIME relative to the transaction. | No  |
| PERMISSION_EFFECTIVE_DATE   | Permit effective date. Format is MM/DD/YYYY.   | No  |
| PERMISSION_EXPIRATION_DATE  | Permit expiration date. Format is MM/DD/YYYY.  | No  |
| ANONYMOUS_USER_ID           | An ORACLE generated value to identify Anonymous reservations made on behalf of the customer.<br><br>This parameter is used when the ReservePermissionQuantity Web method is called multiple times by the same session.                           | No  |
| NUMBER_OF_UNITS             | Number of days, uses, or dollar amount of value to place on value credential.  | No, unless permission.number.range.permit.type.physicalgroup.type is value credential and if value credential |

|  |  |  |
|--|--|--|
|  |  | has value<br>credential fee<br>type schedule |
|--|--|--|

### Return XML Parameters

| Parameter              | Description   |
|------------------------|---|
| PERMISSION_UID_LIST    | A comma separated list of permit UUIDs  |
| PERMISSION_NUMBER_LIST | A comma separated list of permit numbers  |
| NUMBER_RESERVED        | The number of permits reserved  |
| ANONYMOUS_USER_ID      | An ORACLE generated value to identify Anonymous reservations made on behalf of the customer (returned when ENTITY_UUID is blank). |
| PERMISSION_PRICE       | Price of the individual permits that were successfully reserved   |
| PERMISSION_TAX_1       | Tax amount #1 due for the individual permits that were successfully reserved  |
| PERMISSION_TAX_2       | Tax amount #2 due for the individual permits that were successfully reserved  |
| PERMISSION_TAX_3       | Tax amount #3 due for the individual permits that were successfully reserved  |
| TOTAL_PERMISSION_PRICE | Total price of the permit quantity including taxes.   |
| TOTAL_PERMISSION_TAX_1 | Total price of tax amount #1 for the permit quantity.   |
| TOTAL_PERMISSION_TAX_2 | Total price of tax amount #2 for the permit quantity.   |
| TOTAL_PERMISSION_TAX_3 | Total price of tax amount #3 for the permit quantity.   |

### Usage Notes

This method will reserve one or more permits from the specified number range and control group.

- All anonymous reservations must be sold using the new SellPermissionQuantity Web method. This method will accept a blank ENTITY\_UUID to perform an anonymous reservation.
- For permits with a “Set at Time of Sale” effective and/or expiration date, the permit effective and/or expiration date must be passed in order to receive the correct price.
- Permits can only be reserved from a single permit control group and for the same Effective/Expiration date range.

## ReturnTerminatePermission

This Web method allows users to return and/or terminate permits. A new privilege, “Web Services Permits: Return/Terminate Permit” will be added to T2 Flex and is required to access the web method.

### Input XML Parameters

| Input                 | Description  | Required?         |
|-----------------------|--|-------------------|
| TerminatePermission   | Use a 1 or 0 to indicate whether the permit should be returned (0) or terminated (1).  | Yes               |
| PermissionUID         | The UID of the permission to return/terminate.   | Yes               |
| CashDrawerUID         | The web service cash drawer UID for the payment.   | Yes               |
| ReturnReason          | The UID corresponding to the reason for return (1=Returned for Refund, 2=Return for Upgrade, 3=Return for Exchange).   | Yes               |
| ReturnDate            | Date of the permit return in the format YYYY-MM-DD.  | Yes               |
| IncludeDeposit        | Use a 1 or 0 to indicate whether or not the deposit should also be refunded.   | Yes               |
| RefundCalculatedValue | Use a 1 or 0 to indicate whether or not the web method should refund using the calculated value.   | Yes               |
| RefundAmount          | The amount to refund (allows two decimals). This value is ignored if “RefundCalculatedValue” is true (1).  | No                |
| DistributeType        | Only applies to permits on payment plans.<br>0=Distribute refund evenly across all payments,<br>1=Distribute refunds to last payments first. This option is valid for permits on both internal and external payment plans. | No, defaults to 0 |

### Returned XML Parameters

| Parameter Name                  | Description  |
|---------------------------------|--|
| ReturnTerminatePermissionResult | Returns the UID in a string for the successfully returned/terminated permit, the return/terminate date, and the return/terminate amount. |

### Usage Notes

When returning permits under strict inventory control using web services, custody is automatically passed back to main inventory and is not given to the web user.

This web method does not use primitive types. Rather, the information is encapsulated in a special wrapper object (complex type). Users will not be able to use the web method from the discovery page but will instead need to use a suitable test program such as *soapUI*.

## SellPermission() v1.4

### Input XML Parameters

| Parameter                 | Description  | Required  |
|---------------------------|--|---|
| PERMISSION_UID            | Permit UID (PERMISSION.PER_UID) to sell for active, issued, or expired permits | Yes for inventoried number ranges. Optional for non-inventoried ranges.                   |
| ENTITY_UID                | Customer UID (ENTITY.ENT_UID)  | Yes   |
| CASH_DRAWER_UID           | Cash Drawer UID (CASHDRAWER_LKP.CDL_UID)                                       | Yes   |
| CONTROL_GROUP_UID         | Control Group UID (PERMISSION_CONTROL_GROUP.PEC_UID)                           | Yes   |
| STALL_ID                  | Stall ID   | No  |
| SPACE_TYPE_UID            | Space Type UID (PARKING_STALL_TYPE_LKP.PST_UID)                                | No  |
| POSSESSION_REQUIRED       | Is possession confirmation required? 1 = true, 0 = false (default)             | No  |
| MAILING_REQUIRED          | Does customer request that the permit is mailed? 1 = true, 0 = false (default) | No  |
| LINKED_VEHICLE_UID        | Vehicle UID (VEHICLE.VEH_UID) to link to permit                                | No  |
| LINKED_PROPERTY_UID       | Property UID (PROPERTY.PRO_UID) to link to permit                              | No  |
| AMOUNT_PAID               | Amount paid by customer including tax  | Yes   |
| PAYMENT_TYPE_UID          | Payment Method UID (PAYMENT_METHOD_MLKP.PMM_UID)                               | Yes   |
| REFERENCE_NOTE            | Reference Note on payment  | No  |
| PERMISSION_EFFECTIVE_DATE | Format is MM/DD/YYYY   | Yes, if selling an Event permit or if "Set at Point of Sale" is selected in control group |

|                                |  |  |
|--------------------------------|--|--|
|                                |  | configuration.   |
| PERMISSION_EXPIRATION_DATE     | Format is MM/DD/YYYY   | Yes, if selling an Event permit or if “Set at Point of Sale” is selected in control group configuration. |
| FORCE_OVERRIDE_EXPIRATION_DATE | Allows users to override the permit expiration date as part of the online permit sale                        | No, but selecting requires PERMISSION_EXPIRATION_DATE  |
| SHIPPING_METHOD_UID            | UID of shipping method (SHIPPING_METHOD_MLKP.SHP_UID) to be used for mailing the permission.                 | Yes, if mailing required.  |
| MAILING_ADDRESS_UID            | Address UID (COR_ADDRESS.COR_UID) of where to mail a permit  | Yes, if mailing required.  |
| PAYPLAN_TRANSFER_AGENT_UID     | Payment Plan Transfer Agent UID (TRANSFER_AGENCY_MKLP.TAL_UID)   | Yes, if a pay method of external pay plan is used.   |
| PAYPLAN_TEMPLATE_UID           | Payment Plan Template UID (PAYMENT_SCHEDULE_TEMP_MLKP.PSE_UID)   | No   |
| PAYPLAN_START_DATE             | Payment Plan Start Date<br>Format is MM/DD/YYYY  | No, if pay plan used and no date specified the current date will be used                                 |
| PAYPLAN_TIME_PERIOD_UID        | Payment plan time period UID (PAYMENT_PLAN_PERIOD_LKP.PPP_UID). Values: 1-Weekly, 2-Semi-Monthly, 3- Monthly | Yes, if pay plan used and template not specified.  |
| PAYPLAN_TAX_TYPE               | Payment plan tax type UID (PAY_PLAN_TAX_STATUS_LKP.PAT_UID). Values: 2- Pre-Tax, 3- Post-Tax                 | Yes, if an external pay plan used and required by transfer agent.  |
| PAYPLAN_NUM_PAYMENTS           | Number of payment plan payments. Must be a whole number.   | Yes, if pay plan used, template not specified, and amount per payment not specified.                     |

|                             |  |  |
|-----------------------------|--|--|
| PAYPLAN_AMOUNT_PER_PAYMENT  | Amount of each payment plan payment.   | Yes, if pay plan used, template not specified, and number of payments not specified.   |
| EMAIL_NOTIFICATION_REQUIRED | Is e-mail notification of permit shipment requested? 1 = true, 0 – false (default)   | No   |
| EMAIL_UID                   | E-mail address UID (COR_EMAIL.COE_UID) to be used for shipment notification  | Yes, if e-mail notification of shipment is requested.  |
| PERMISSION_SALE_PRICE       | Permit sale price. Use if the sale price differs from the PERMISSION_PRICE returned by ReservePermission or GetPermissionPrice method.                         | No   |
| VALID_LOCATIONS             | XML formatted string containing facilities, areas, stall types, and stalls where a permit is valid (see further explanation at the beginning of this section). | No   |
| ACCOUNT_CREDIT_LIST         | A comma separated list of Account Credits (ACCOUNT_CREDITS.ACR_UID)  | Yes, if account payment method used.   |
| ALLOC_FACILITY_UID          | The UID corresponding to the allotment-controlled facility selected for the sale.  | Yes, if permit sale is allotment-controlled and has an allotment type of facility.   |
| ALLOC_FACILITY_GROUP_UID    | UID corresponding to the allotment-controlled facility group selected for the sale   | Yes, if permit sale is allotment-controlled and has an allotment type of facility group.   |
| NUMBER_OF_UNITS             | Number of days, uses, or dollar amount of value to place on value credentials  | No, unless permission.number.range.permit.type.physicalgroup.type is value credential and if value credential has value credential |

|                     |  |   |
|---------------------|--|---|
|                     |  | fee type schedule   |
| BillOnUseThirdParty | Third party to bill for additional uses                      | No, only used if permission.number range.permit type.physicalgroup type is value credential |
| THIRD_PARTY_UID     | UID of a valid third party to be used as the payment method. | No  |

### Return XML Parameters

| Parameter      | Description                              |
|----------------|--|
| PERMISSION_UID | UID of permit that was successfully sold |
| RECEIPT_UID    | UID of the receipt for this payment      |

### Usage Notes

This method is used to sell a permit after it has first been reserved. It uses the fee schedule of type value credential for a value credential permit passed in to determine the permit's price.

Permits can be sold with a partial amount, leaving the remaining balance on the permit. The Web method can be called multiple times on the same permit to reduce a balance to \$0. A new receipt is generated for each payment.

For each inventoried permit to be sold, this method will perform the following steps:

- Verify that the permit has been reserved for the customer specified in the ENTITY\_UID parameter.
- Check any permit district restrictions on the control group.
 

**Note:** If the **Residential Permit Sales Based On Property** configuration setting is enabled, identify the property to assign to the permit. This property should have a permit district that meets the criteria defined by the control group.
- Verify that the customer can purchase the reserved permit.\*
- Check that the permit sale does not violate Permit Allotment rules.
- Check that the cash drawer specified by CASH\_DRAWER\_UID is open, and if it is not, open it.\*
- Sell the permit.\*

\*Only these criteria are checked for non-inventoried permits. In addition, the following steps are performed for non-inventoried permits:

- Validate that the permit can be sold through Web Services.
- Create and populate/reserve the non-inventoried permit.

- Validate the existing business rules for the created permit.
- Validate that the Valid Locations parameter meets the appropriate business rules including Facility Tier requirements, if applicable.

Currently this method cannot use the following payment types:

- Account Deposit
- Import Payment
- Payment Plan Refund

Currently this method cannot use the following permit types:

- Non-tracked
- Printable (unless the **Allow Sales from Web Services Cash Drawers** and **Do Not Print Permits** settings are enabled).
- Non-reserveable

When selling linkable permits, the linkable permit must be sold first, and then the audit permit may be sold. If the customer does not have an active linkable permit but requires it, the Web method will not allow the sale and will generate an error.

Note the following properties of permit sales using an internal or external payment plan for the payment type:

- The transfer agency is only required for external payment plans, but any associated custom fields are not required.
- For external payment plans, if a payment plan template UID is specified, the payment plan time period, and number of payments or amount per payment are not required. However, if a payment plan template is specified, then any values passed for payment plan time period, number of payments, or the amount per payment will override the values found in the template. Therefore, To use the values on a payment plan template do not send the PAYPLAN\_TIME\_PERIOD\_UID, PAYPLAN\_NUM\_PAYMENTS, or the PAYPLAN\_AMOUNT\_PER\_PAYMENT parameters.
- If a valid payment plan template is not specified, the payment plan time period and either the number of payment or the amount per payment are required. If both the number of payments and the amount per payment are included an error will be returned.

**Note:** A valid payment plan template must have an UID greater than zero (0).

- If the number of payments exceeds the expiration date of the permission, an error will not be returned.

**Note:** In T2 Flex this is an overrideable error, however the Web Method will allow it to occur without a warning.

- The payment plan tax type is only required if **Set Pre/Post Tax at Checkout** is selected on the Transfer Agency setup. If that setting is not enabled, any value sent for payment plan tax type will not be used.



- If the number of payments is specified then the amount per payment will be calculated and is not required. If the amount per payment is specified then the number of payments will be calculated and is not required. If both the number of payments and the amount per payment are specified then an error will be returned.

If a positive value is included for the `PERMISSION_SALE_PRICE` parameter, that value will override the value generated by Flex. If this parameter is not passed, the default price will be used. If a value of zero is passed, a payment method of “No Charge” must be used.

When selling strict inventory controlled permits that are allocated to a Web services cash drawer, the current Web user must have custody.

Your organization may have additional permit eligibility business rules other than those verified by this method. Checking for those additional criteria is the responsibility of the calling program. When this method is called, it will check to determine if any external Eligibility Rules (determined in Permit System Settings) apply to the permit being authorized.

**Note:** If your organization’s business rules conflict with the criteria listed here, do not call this method, and instead implement eligibility checks by querying the T2 Flex database directly from your program.

- If the Eligibility Rule is **External Rules Only**, the method calls the external Single Permit Eligibility Web Method to determine eligibility. If the permit is not eligible, the error returned will be the text value of the `NOT_ELIGIBLE_REASON` parameter from the external method.
- If the Eligibility Rule is **T2 Flex Rules External Override**, then T2 Flex rules are sent to the external Single Permit Eligibility Web Method to determine if those eligibility rules are met. If the permit is not eligible, the error returned will be based on the XML return parameters of the external method.

**Note:** See [Chapter 4](#) for more detail on requirements for the external permit eligibility Web Methods.

## SellPermissionQuantity() v1.2

### Input XML Parameters

| Parameter Name                      | Description  | Required? |
|-------------------------------------|--|-----------|
| <code>PERMISSION_UID_LIST</code>    | A comma separated list of <code>PER_UIDs</code> ( <code>PERMISSION.PER_UID</code> )  | Yes       |
| <code>CONTROL_GROUP_UID_LIST</code> | A comma separated list of <code>CONTROL_GROUP_UIDs</code> ( <code>PERMISSION_CONTROL_GROUP.PEC_UID</code> ) index corresponding to the <code>PERMISSION_UID_LIST</code> if more than one CONTROL GROUP UID is specified, otherwise the | Yes       |

|                       |  |                             |
|-----------------------|--|-----------------------------|
|                       | permit UUIDs are assumed to come from the single specified CONTROL_GROUP_UUID  |                             |
| ANONYMOUS_USER_ID     | The ORACLE generated value returned from ReservePermissionQuantity when an Anonymous Reservation process was executed. | No                          |
| ENTITY_UUID           | Customer UUID (ENTITY.ENT_UUID)  | Yes                         |
| CASH_DRAWER_UUID      | Cash Drawer UUID (CASHDRAWER_LKP.CDL_UUID)   | Yes                         |
| AMOUNT_PAID           | Total amount paid by the customer including tax  | Yes                         |
| PAYMENT_TYPE_UUID     | Payment Method UUID (PAYMENT_METHOD_MLKP.PM_M_UUID)  | Yes                         |
| POSSESSION_REQUIRED   | Is possession confirmation required? 1 = true, 0 = false (default)   | No                          |
| MAILING_REQUIRED      | Does customer request that the permit is mailed? 1 = true, 0 = false (default)   | No                          |
| LINKED_VEHICLE_UUID   | Vehicle UUID (VEHICLE.VEH_UUID) to link with permissions   | No                          |
| LINKED_PROPERTY_UUID  | Property UUID (PROPERTY.PRO_UUID) to link with permissions   | No                          |
| REFERENCE_NOTE        | Reference note on payment  | No                          |
| SHIPPING_METHOD_UUID  | UUID of shipping method (SHIPPING_METHOD_MLKP.SHIP_UUID) to be used for mailing the permission                         | Yes, if mailing is required |
| MAILING_ADDRESS_UUID  | Address UUID (COR_ADDRESS.COR_UUID) of where to mail a permit  | Yes, if mailing is required |
| EMAIL_NOTIFY_REQUIRED | Is e-mail notification of permit shipment requested? 1 = true, 0 = false (default)                                     | No                          |

|                                 |   |   |
|---------------------------------|---|---|
| EMAIL_UID                       | E-mail address UID (COR_EMAIL.COE_UID) to be used for shipment notification   | Yes, if email notification is requested   |
| STALL_ID_LIST                   | A comma separated list of Stall IDs corresponding to the PERMISSION_UID_LIST  | No  |
| SPACE_TYPE_UID_LIST             | A comma separated list of SPACE_TYPE_UIDs corresponding to the PERMISSION_UID_LIST  | No  |
| PERMISSION_EFFECTIVE_DATE_LIST  | A comma separated list of Permit Effective dates corresponding to the PERMISSION_UID_LIST   | No  |
| PERMISSION_EXPIRATION_DATE_LIST | A comma separated list of Permit Expiration dates index corresponding to the PERMISSION_UID_LIST  | No  |
| VALID_LOCATIONS                 | XML formatted string containing facilities, areas, stall types, and stalls where the permits are valid (see further explanation at the beginning of this section) | No  |
| ACCOUNT_CREDIT_LIST             | A comma separated list of Account Credits (ACCOUNT_CREDITS.ACR_UID)   | Yes, if account payment method used   |
| ALLOC_FACILITY_UID              | The UID corresponding to the allotment-controlled facility selected for the sale.   | Yes, if permit sale is allotment-controlled and has an allotment type of facility.                                      |
| ALLOC_FACILITY_GROUP_UID        | The UID corresponding to the allotment-controlled facility group selected for the sale  | Yes, if permit sale is allotment-controlled and has an allotment type of facility group.                                |
| NUMBER_OF_UNITS                 | Number of days, uses, or dollar amount of value to place on value credential  | No, unless permission.number.range.permit.type.physicalgroup.type is value credential and if value credential has value |

|                     |  |   |
|---------------------|--|---|
|                     |  | credential fee type schedule  |
| BillOnUseThirdParty | Third party to bill for additional uses                      | No, only used if permission.number.range.permit.type.physicalgroup.type is value credential |
| THIRD_PARTY_UID     | UID of a valid third party to be used as the payment method. | No  |

### Return XML Parameters

| Parameter Name      | Description                                      |
|---------------------|--|
| PERMISSION_UID_LIST | UIDs of the permits that were successfully sold. |
| RECEIPT_UID         | UID of the related receipt.                      |

### Usage Notes

This web method will sell one or more permits to a specified ENTITY\_UID who has previously reserved the permits either by ENTITY or ANONYMOUS\_USER\_ID.

- When more than one value is specified for a “\_LIST” parameter, a value must be provided for each permit in the PERMISSION\_UID\_LIST.
- When a single value is specified for a “\_LIST” parameter, the value will be applied to each permit in the PERMISSION\_UID\_LIST.
- When selling strict inventory controlled permits that are allocated to a Web services cash drawer, the current Web user must have custody.
- Currently this method cannot use the following payment types:
  - Account
  - Account Deposit
  - Import Payment
  - Payment Plan
  - Payment Plan Refund
- Currently this method cannot use the following permit types:
  - Non-tracked
  - Printable
  - Non-Reserveable
- When selling linkable permits in quantity, the linkable permit and the audit permit may be sold at the same time, but the linkable permit must be first in the list of permits. If the customer does not have an active linkable permit but requires it, the Web method will not allow the sale and will generate an error.

**SetLostStolen() v1.0****Input XML Parameters**

| Parameter          | Description   | Required |
|--------------------|---|----------|
| PERMISSION_UID     | Permit UID (PERMISSION.PER_UID)                           | Yes      |
| MISSING_REASON_UID | Missing Reason UID<br>(PER_MISSING_REASON_LKP.PMR_UID)    | No       |
| GENERATE_LETTER    | Generate Letter? 1 = true, 0 = false (default)            | No       |
| LETTER_ADDRESS_UID | Address UID (COR_ADDRESS.COR_UID)<br>for generated letter | No       |
| LETTER_EMAIL_UID   | Email UID (COR_EMAIL.COE_UID) for<br>generated letter     | No       |

**Return XML Parameters**

| Parameter        | Description                     |
|------------------|---------------------------------|
| PERMISSION_UID   | UID of permit that was updated  |
| DEACTIVATION_FEE | Deactivation fee amount, if any |
| LETTER_UID       | Letter UID                      |

**Usage Notes**

This method is used to set the status of an active permit to Lost/Stolen. If a deactivation fee is required for a status change it will be added to the permission. If the generate letter parameter is set to true, an address UID or email address UID is required input. The method will look up the customer currently related to the permission, and return an error if the specified address or email address is not related to that entity.

**SetPermissionActiveAlternateID() v1.0****Input XML Parameters**

| Parameter              | Description                           | Required |
|------------------------|---------------------------------------|----------|
| AlternateIDNumber      | Alt ID number                         | Yes      |
| AlternateIDDescription | Alt ID field description from T2 Flex | Yes      |

**Return XML Parameters**

| Parameter                | Description                           |
|--------------------------|---------------------------------------|
| ALTERNATE_ID_DESCRIPTION | Alt ID field description from T2 Flex |
| ALTERNATE_ID_NUMBER      | Alt ID number                         |
| ALTERNATE_ID_ACTIVE      | True or false if Alt ID is active     |

**SetSuspendRevoke() v1.0****Input XML Parameters**

| Parameter             | Description   | Required |
|-----------------------|---|----------|
| PERMISSION_UID        | Permit UID (PERMISSION.PER_UID)                           | Yes      |
| DEACTIVATE_REASON_UID | Deactivate Reason UID (PER_DEACTIVATE_REASON_LKP.PDK_UID) | No       |
| GENERATE_LETTER       | Generate Letter? 1 = true, 0 = false (default)            | No       |
| LETTER_ADDRESS_UID    | Address UID (COR_ADDRESS.COR_UID) for generated letter    | No       |
| LETTER_EMAIL_UID      | Email UID (COR_EMAIL.COE_UID) for generated letter        | No       |

**Return XML Parameters**

| Parameter        | Description                      |
|------------------|----------------------------------|
| PERMISSION_UID   | UID of permit that was updated.  |
| DEACTIVATION_FEE | Deactivation fee amount, if any. |
| LETTER_UID       | Letter UID                       |

**Usage Notes**

This method is used to set the status of an active permit to Inactive. If a deactivation fee is required for a status change it will be added to the permission. If the generate letter parameter is set to true, an address UID or email address UID is required input. The method will look up the customer currently related to the permission, and return an error if the specified address or email address is not related to that entity.

**UnreservePermission() v1.0****Input XML Parameters**

| Parameter      | Description                                     | Required |
|----------------|---|----------|
| PERMISSION_UID | UID (PERMISSION.PER_UID) of permit to unreserve | Yes      |

**Return XML Parameters**

| Parameter      | Description                                    |
|----------------|--|
| PERMISSION_UID | UID of permit that was successfully unreserved |

**Usage Notes**

This method is used to unreserve a permit that was previously reserved.

## UnreservePermissionQuantity() v1.0

### Input XML Parameters

| Parameter Name      | Description   | Required?                                |
|---------------------|---|--|
| PERMISSION_UID_LIST | Comma delimited list of permit<br>UIDs to unreserve.  | Yes                                      |
| ENTITY_UID          | Customer UID<br>(ENTITY.ENT_UID)  | Yes, if<br>ANONYMOUS_USER_ID<br>is EMPTY |
| ANONYMOUS_USER_ID   | An ORACLE generated value to<br>identify Anonymous reservations<br>made on behalf of the customer.<br><br>This parameter is used when the<br>ReservePermissionQuantity web<br>method is called multiple times<br>by same session. | Yes, if ENTITY_UID is<br>EMPTY           |

### Returned XML Parameters

| Parameter Name      | Description   |
|---------------------|---|
| PERMISSION_UID_LIST | Comma delimited list of permit UIDs that were successfully<br>unreserved. |

### Usage Notes

This method will unreserve one or permits for the specified customer or anonymous user who previously reserved the permits.

- This web method can be called in conjunction with the original ReservePermission web method.
- If the PER\_UIDs in the PERMISSION\_UID\_LIST include both Anonymous and Regular reservations, the ENTITY\_UID and the ANONYMOUS\_USER\_ID should both be passed to unreserve the permits.

## UpdatePermissionToProfileRelationship() v1.0

### Input XML Parameters

| Parameter Name                      | Description  | Required? |
|-------------------------------------|--|-----------|
| PERMISSION_PROFILE_RELATIONSHIP_UID | The Permission to<br>Credit Card Profile<br>Relationship to be<br>edited.          | Yes       |
| PERMIT_UID                          | The permission that the<br>Permission to Credit<br>Card Profile<br>Relationship is | Yes       |

|           |  |    |
|-----------|--|----|
|           | associated with.<br>Note: this is used to ensure the Relationship being edited is for the permission specified.              |    |
| STARTDATE | The starting date for the relationship.<br>Note: when this is not provided the StartDate will be stored as the current date. | No |
| ENDDATE   | The ending date for the relationship.  | No |

**Returned XML Parameters**

| Parameter Name                      | Description   |
|-------------------------------------|---|
| PERMISSION_PROFILE_RELATIONSHIP_UID | The Permission to Credit Card Profile Relationship that was edited. |

**Usage Notes**

This web method will be called to update an existing Permission Credit Card Profile Relationship in the T2 Flex database.

**UpdateShippedPermission() v1.0****Input XML Parameters**

| Parameter            | Description  | Required |
|----------------------|--|----------|
| PERMISSION_UID       | Permit UID (PERMISSION.PER_UID) of the shipped permit.         | Yes      |
| MAILING_DATE         | Date the permit was mailed. Format is MM/DD/YYYY.              | Yes      |
| MAIL_TRACKING_NUMBER | The tracking number for the mailed permit (UPS/FedEx numbers). | No       |

**Return XML Parameters**

| Parameter      | Description                                 |
|----------------|---|
| PERMISSION_UID | Permit UID that was successfully "shipped." |

**Usage Notes**

This method is used to update a permit for which mailing was requested, to indicate that it has been shipped. This method can only be used once per permission.

The following process is used to mark a permit as having been shipped:



- The permit is validated for its existence.
- The method checks that mailing was requested for the permit and that the permit does not already have a mailing date.
- Mailing date and tracking number (if any) are set on the permission.

## T2\_FLEX\_POSSTATIONS

This Web service and its methods are for internal T2 Systems use only and should not be used by customers.

## T2\_FLEX\_PROPERTIES

### InsertUpdateProperty() v1.0

#### Input XML Parameters

| Parameter                | Description  | Required?      |
|--------------------------|--|----------------|
| PROPERTY_UID             | The property to update.  | Yes, if update |
| PROPERTY_NAME            | The name of the property.  | Yes            |
| STREET_NUMBER            | The street number of the property.                               | Yes            |
| STREET_NAME              | The street name of the property.                                 | Yes            |
| SUITE_APARTMENT          | The suite/apartment of the property.                             | No             |
| CITY                     | The city where the property is located.                          | No             |
| PROPERTY_TYPE_UID        | The property type the property belongs under.                    | Yes            |
| PERMIT_DISTRICT_UID      | The permit district where the property is located.               | No             |
| UNITS                    | The number of units at the property.                             | No             |
| OVERRIDE_MAXIMUM_PERMITS | The maximum permits to be stored on the property as an override. | No             |

#### Return XML Parameters

| Parameter    | Description                             |
|--------------|---|
| PROPERTY_UID | The UID of the new or updated property. |

#### Usage Notes

- The property style will be derived from the property type.

- The maximum permits calculation will be made based on information derived from the property type and the number of units at the property.
- If the permit district field is not provided, the method will attempt to identify the permit district by comparing the property to records in the Property Address Range lookup. An error will be returned if the permit district cannot be automatically identified.
- If the units' field is not provided, the method will set the field to be equal to 1.

## T2\_FLEX\_REPORTING

### ExecuteReport() v1.0

#### Input XML Parameters

| Parameter Name    | Description   | Required? |
|-------------------|---|-----------|
| REPORT_PARAMETERS | XML-formatted string containing variable report parameters to be passed to the report. See example below. | X         |
| REPORT_UID        | The UID of the report template to be executed   | X         |
| TARGET_OBJ_TYPE   | Type of object to which the resulting document will be attached.  |           |
| TARGET_OBJ_UID    | UID of object to which the resulting document will be attached  |           |
| NOTE_TYPE         | UID of Note Type Lookup, required if TARGET_OBJ_TYPE and TARGET_OBJ_UID is specified.                     |           |
| NOTE_UID          | UID of pre-existing Note to which the resulting document will be attached                                 |           |

#### Output XML Parameters

| Parameter Name  | Description   |
|-----------------|---|
| DOCUMENT_UID    | UID of the created Document (if NOTE_UID or TARGET_OBJ_TYPE and TARGET_OBJ_UID are specified)                     |
| BINARY_DOCUMENT | Base64-encoded string of document (if NOTE_UID, TARGET_OBJ_TYPE and TARGET_OBJ_UID parameters are omitted).       |
| NOTE_UID        | UID of Note to which the Document was attached (if NOTE_UID or TARGET_OBJ_TYPE and TARGET_OBJ_UID are specified). |

#### Usage Notes

This method executes a T2 Flex Report (document), either returning the report contents as a string or saving the report as an attachment to a note.

## GetDocument() v1.0

### Input XML Parameters

| Parameter Name | Description                             | Required? |
|----------------|---|-----------|
| DOCUMENT_UID   | The UID of the Document to be retrieved | Yes       |

### Output XML Parameters

| Parameter Name  | Description                                 |
|-----------------|---|
| BINARY_DOCUMENT | Base64-encoding of the Document content     |
| DOCUMENT_TYPE   | MimeType of Document (from Document object) |
| DOCUMENT_NAME   | Filename of Document as stored in Flex      |
| DOCUMENT_UID    | UID of the retrieved Document               |

### Usage Notes

This method retrieves report (document) details and content from T2 Flex. Content will need to be converted back into byte form (from Base64) before it is usable.

## InsertDocument() v1.0

This method attaches a Document (based on a Base64-encoded string representing file contents) to an existing or new note record in T2 Flex.

### Input XML Parameters

| Parameter Name  | Description   | Required? |
|-----------------|---|-----------|
| BINARY_DOCUMENT | Base64-encoded string of Document contents  | Yes       |
| TARGET_OBJ_TYPE | Type of object to which the resulting document will be attached                       | No        |
| TARGET_OBJ_UID  | UID of object to which the resulting document will be attached                        | No        |
| NOTE_TYPE       | UID of Note Type Lookup, required if TARGET_OBJ_TYPE and TARGET_OBJ_UID is specified. | No        |
| NOTE_UID        | UID of Note to which the resulting document will be attached                          | No        |
| DOCUMENT_NAME   | Name of the document to be stored   | Yes       |

### Output XML Parameters

| Parameter Name | Description                                    |
|----------------|--|
| DOCUMENT_UID   | UID of new Document                            |
| NOTE_UID       | UID of Note to which the Document was attached |

## T2\_FLEX\_RESERVATION

### CreateEventFacilityReservation() v1.0

This method accepts reservation details and creates a PermitNow reservation in T2 Flex. For ARC reservations, use the CreateReservation() Web method.

#### Input XML Parameters

| Parameter                 | Description   | Required |
|---------------------------|---|----------|
| EventFacilityUid          | UID of the event facility the reservation is associated with.                                       | Yes      |
| EventValueUid             | UID of the event value charged for the reservation.   | Yes      |
| ThirdPartyUid             | If ValidationControlGroupUid is specified, the responsible third party UID is required.             | No       |
| ReservationDescription    | Description of the reservation.   | Yes      |
| RequestorFirstName        | First name of the customer requesting the reservation.  | No       |
| RequestorLastName         | Last name of the customer requesting the reservation.   | No       |
| ReservationStartDate      | Start date of the reservation (must be within start/end dates of the event facility).               | Yes      |
| ReservationEndDate        | Start date of the reservation (must be within start/end dates of the event facility).               | Yes      |
| DailyUses                 | Number of parkers that can enter on this reservation per day. 1 or greater.                         | Yes      |
| CanExceedDailyUses        | Whether or not the number of daily uses can be exceeded. 1 or 0 for yes or no.                      | No       |
| ValidationControlGroupUid | UID of the validation control group associated to the third party responsible for this reservation. | No       |
| CollectNames              | Whether or not field user is required to collect names as parkers enter. 1 or 0 for yes or no.      | No       |
| ChargeForUnusedValue      | Ignore—not currently used.  | No       |
| ReservationUsers          | Contains a ReservationName complex type* parameter for each item on appeal (see next item).         | No       |
| ReservationName           | Complex type* collection of the following parameters used to name specific reservation users:       | No       |

|  |   |  |
|--|---|--|
|  | <ul style="list-style-type: none"> <li>• FirstName</li> <li>• MiddleName</li> <li>• LastName</li> </ul> |  |
|--|---|--|

### Return XML Parameters

| Parameter                            | Description  |
|--------------------------------------|--|
| CreateEventFacilityReservationResult | A string containing the UID of the inserted reservation. |

### Usage Notes

\*Complex Type – This type of parameter contains nested parameters, as indicated in the table.

## CreateReservation() v1.0

This method accepts reservation details and creates an ARC reservation in T2 Flex. For PermitNow reservations, use the CreateEventFacilityReservation() Web method.

### Input XML Parameters

| Parameter                    | Description   | Required?                                |
|------------------------------|---|--|
| DESCRIPTION                  | The reservation description.  | Yes                                      |
| RESERVATION_TYPE_UID         | The reservation type. 1 = standard reservation, 2 = PermitNow reservation | Yes                                      |
| REQUESTER_FIRST_NAME         | The requester first name for the reservation.                             | Yes                                      |
| REQUESTER_LAST_NAME          | The requester last name for the reservation.                              | Yes                                      |
| START_DATE                   | The start date for the reservation.                                       | Yes                                      |
| END_DATE                     | The end date for the reservation.   | Yes                                      |
| PERM_CONTROL_GROUP_UID       | The pay on entry permit control group for the reservation.                | Yes, if not linked to a PermitNow event. |
| PARKING_RATE_TYPE_UID        | The parking rate type for the reservation.                                | Yes, if linked to a PermitNow event.     |
| VALIDATION_CONTROL_GROUP_UID | The validation control group for the reservation.                         | No                                       |
| THIRD_PARTY_UID              | The third party to link the reservation.                                  | Yes, if validation control               |

|                              |   |   |
|------------------------------|---|---|
|                              |   | group provided.   |
| FACILITY_UID                 | The facility where the reservation will be used. This is defined automatically if the reservation is linked to a PermitNow event.             | Yes, not required and ignored if a PermitNow event is provided. |
| PERMITNOW_EVENT_UID          | The PermitNow event where the reservation will be used.   | No  |
| VALUE_TYPE_UID               | The value type to be used by the reservation.   | Yes   |
| SET_VALUE                    | The maximum dollar amount or maximum number of uses for the reservation based on the VALUE_TYPE_UID provided.                                 | Yes   |
| CAN_EXCEED_SET_VALUE         | Boolean indicating if the reservation set value can be exceeded.  | Yes   |
| CHARGE_FOR_ALL_UNUSED_SPACES | Determines if the third party will be charged for unused spaces if the VALUE_TYPE_UID is the maximum uses type.                               | Yes   |
| OCCUPANCY_TYPE_UID           | The occupancy type to be used by the reservation uses.  | Yes, if linked to a PermitNow event that tracks occupancy.      |
| COLLECT_NAMES                | Boolean indicating if the cashier station or PermitNow handheld software should collect names on use in the field.                            | No  |
| USER_LIST                    | The list of users who will consume spaces from the reservation. Delimit with a : character, for example "Blankenship, Jerrel:Jenkins, Bob R." | No  |

### Return XML Parameters

| Parameter       | Description                             |
|-----------------|---|
| RESERVATION_UID | The reservation UID for the new record. |

### Usage Notes

The occupancy and sales capacity in Flex are verified when the reservation is inserted. By default, reservations submitted via this method are inserted with a status of “Pending Approval”.

Users can pass in values for reservation custom fields with this method.

## T2\_FLEX\_THIRDPARTY

### GetThirdPartyInvoices

Returns all active unpaid invoices for the third party. This is similar to the Validation Provider > GetValidationProviderInvoices() Web method only it does not require the external system user name and password.

#### Input XML Parameters

| Parameter     | Description   | Required |
|---------------|---|----------|
| ThirdPartyUid | UID of the third party for which you are retrieving an invoice. | Yes      |

#### Return XML Parameters

| Parameter                   | Description  |
|-----------------------------|--|
| GetThirdPartyInvoicesResult | A string including the UID, amount due, and date for each invoice. |

### PayThirdPartyInvoice

This web method allows a specific invoice to be paid. This is similar to the Validation Provider > PayValidationProviderInvoice() Web method only it does not require the external system user name and password.

#### Input XML Parameters

| Parameter                                  | Description   | Required |
|--|---|----------|
| InvoiceUid                                 | UID of the invoice to be paid.  | Yes      |
| EntityUid                                  | UID of the responsible customer.  | No       |
| PaymentMethodUid                           | UID of the payment method.  | Yes      |
| AccountCredits/<br>AccountCreditsListParam | Complex type* collection of the following parameters: <ul style="list-style-type: none"> <li>AccountCreditsUid</li> </ul> | No       |
| PaymentAmount                              | Amount of the payment.  | Yes      |
| CashDrawerUid                              | UID of the Web drawer.  | Yes      |
| ReferenceNote                              | Reference note for the payment.   | No       |

**Return XML Parameters**

| Parameter                  | Description  |
|----------------------------|--|
| PayThirdPartyInvoiceResult | A string including the receipt UID, third party amount due, third party amount invoiced, and third party amount un-invoiced. |

**Usage Notes**

\*Complex Type – This type of parameter contains nested parameters, as indicated in the table.

**T2\_FLEX\_VALIDATIONPROVIDERS****AddValidationProviderUser**

This Web method allows "external system" type user accounts to be created within T2 Flex.

**Input XML Parameters**

| Parameter                   | Description  | Required |
|-----------------------------|--|----------|
| ExternalUserAccountName     | 6 digit user name of the external validation provider user with the Validation Provider User Management privilege.   | Yes      |
| ExternalUserAccountPassword | Password of the external validation provider user.   | Yes      |
| UserDescription             | Description of new external user.  | No       |
| Password                    | Password of new external user.   | Yes      |
| UserPrivileges              | UIDs of privileges to be assigned:<br>IssuePaperlessValidation = 350<br>GetValidationProviderBalance = 351<br>PayInvoice = 353<br>ValidationProviderUserManagement = 354<br>IssuePrintableValidation = 358 | No       |
| UserRoles                   | UIDs of roles to be assigned:<br>ValidationProviderAdmin = 2<br>PrintableValidationAdmin = 3   | No       |



**Return XML Parameters**

| Parameter                       | Description  |
|---------------------------------|--|
| AddValidationProviderUserResult | A string indicating whether or not the insert was successful that contains the user account UID, user name, and description. |

**AvailableValidationControlGroupsForValidationProvider**

Returns a list of active validation control groups that are valid and available for use by a validation provider (not over the debt threshold).

**Input XML Parameters**

| Parameter                   | Description   | Required |
|-----------------------------|---|----------|
| ExternalUserAccountName     | 6 digit user name of the external validation provider user. | Yes      |
| ExternalUserAccountPassword | Password of the external validation provider user.          | Yes      |

**Return XML Parameters**

| Parameter   | Description   |
|---|---|
| AvailableValidationControlGroupsForValidationProviderResult | A string including a list of available validation control groups, each containing the following: UID, description, discount type UID, discount type description, discount amount. |

**Usage Notes**

The Validation Provider is determined via the External System User Account.

**GetAssignedBooleanUserPrivilegesForExternalSystemUser**

Returns a list of privileges assigned to the given external system user.

**Input XML Parameters**

| Parameter                   | Description   | Required |
|-----------------------------|---|----------|
| ExternalUserAccountName     | 6 digit user name of the external validation provider user. | Yes      |
| ExternalUserAccountPassword | Password of the external validation provider user.          | Yes      |

**Return XML Parameters**

| Parameter   | Description  |
|---|--|
| GetAssignedBooleanUserPrivilegesForExternalSystemUserResult | A string containing the following information for the external system user: validation provider UID, external system user UID, with a list of assigned privileges per user UID (privilege UID and privilege type description). |

**GetValidationProviderBalance**

Returns the balances associated to the validation provider.

**Input XML Parameters**

| Parameter                   | Description  | Required |
|-----------------------------|--|----------|
| ExternalUserAccountName     | 6 digit user name of the external validation provider user with the Get Validation Provider Balance privilege. | Yes      |
| ExternalUserAccountPassword | Password of the external validation provider user.   | Yes      |

**Return XML Parameters**

| Parameter                          | Description  |
|------------------------------------|--|
| GetValidationProviderBalanceResult | A string containing the following information for the given validation provider: responsible entity UID and name, third party UID and name, pre-pay balance, validation hold amount, third party amount due, third party un-invoiced amount. |

**Usage Notes**

The Validation Provider is determined via the External System User Account.

**GetValidationProviderInvoices**

Returns all active unpaid invoices for the third party associated with the validation provider.

**Input XML Parameters**

| Parameter                   | Description  | Required |
|-----------------------------|--|----------|
| ExternalUserAccountName     | 6 digit user name of the external validation provider user with the Pay Invoice privilege. | Yes      |
| ExternalUserAccountPassword | Password of the external validation provider user.   | Yes      |

**Return XML Parameters**

| Parameter                           | Description  |
|-------------------------------------|--|
| GetValidationProviderInvoicesResult | A string including the UID, amount due, and date for each invoice. |

**Usage Notes**

The Validation Provider is determined via the External System User Account.

**GetValidationProviderUsers**

Returns a list of external system user accounts that are related to the validation provider.

**Input XML Parameters**

| Parameter                   | Description  | Required |
|-----------------------------|--|----------|
| ExternalUserAccountName     | 6 digit user name of the external validation provider user with the Validation Provider User Management privilege. | Yes      |
| ExternalUserAccountPassword | Password of the external validation provider user.   | Yes      |

**Return XML Parameters**

| Parameter                        | Description  |
|----------------------------------|--|
| GetValidationProviderUsersResult | A string including a list of users, their descriptions, and associated roles and privileges. |

**Usage Notes**

The Validation Provider is determined via the External System User Account.

**IssueOnlineCoupon**

Creates a sell-on-use validation. This validation will not be associated with a permit until it is processed.

**Input XML Parameters**

| Parameter                   | Description   | Required |
|-----------------------------|---|----------|
| ExternalUserAccountName     | 6 digit user name of the external validation provider user with the Issue Printable Validation privilege. | Yes      |
| ExternalUserAccountPassword | Password of the external validation provider user.  | Yes      |
| ValidationControlGroupUid   | UID of validation control group from which the coupon is issued.  | Yes      |

**Return XML Parameters**

| Parameter               | Description  |
|-------------------------|--|
| IssueOnlineCouponResult | A string including the validation UID, validation number and report template for the coupon. |

**IssueOnlineValidation**

Creates a sell-on-use validation and associates that validation with a given permit.

**Input XML Parameters**

| Parameter                   | Description   | Required |
|-----------------------------|---|----------|
| ExternalUserAccountName     | 6 digit user name of the external validation provider user with the Issue Paperless Validation privilege. | Yes      |
| ExternalUserAccountPassword | Password of the external validation provider user.  | Yes      |
| ValidationControlGroupUid   | UID of validation control group from which the validation is issued.                                      | Yes      |
| PermissionNumber            | UID of validation control group to which the permit is issued.  | Yes      |

**Return XML Parameters**

| Parameter                   | Description   |
|-----------------------------|---|
| IssueOnlineValidationResult | A string including the validation UID, validation number, permit UID and permit number. |

**Usage Notes**

The Validation Provider is determined via the External System User Account.

The Web method checks the Exit Device online availability to see that all DCSs with an Exit Pay Station have been online in the past 'X' minutes. No validations should be issued if this business logic fails.

A sell on use validation is created and linked to the permit via the new relationship so that when the parker leaves, the validation can be accessed and applied.

## ModifyValidationProviderUser

Allows external system type user accounts to be modified.

### Input XML Parameters

| Parameter                   | Description  | Required |
|-----------------------------|--|----------|
| ExternalUserAccountName     | 6 digit user name of the external validation provider user with the Validation Provider User Management privilege.   | Yes      |
| ExternalUserAccountPassword | Password of the external validation provider user.   | Yes      |
| UserName                    | 6 digit user name of the user being modified.  | Yes      |
| Password                    | Password of external user.   | No       |
| UserDescription             | Description of external user.  | No       |
| IsActive                    | 1 or 0 for Yes or No.  | No       |
| UserPrivilegesAssigned      | UIDs of privileges to be assigned:<br>IssuePaperlessValidation = 350<br>GetValidationProviderBalance = 351<br>PayInvoice = 353<br>ValidationProviderUserManagement = 354<br>IssuePrintableValidation = 358 | No       |
| UserPrivilegesRemoved       | UIDs of privileges to be removed.  | No       |
| UserRolesAssigned           | UIDs of roles to be assigned:<br>ValidationProviderAdmin = 2<br>PrintableValidationAdmin = 3   | No       |
| UserRolesRemoved            | UIDs of roles to be removed.   | No       |

**Return XML Parameters**

| Parameter                          | Description   |
|------------------------------------|---|
| ModifyValidationProviderUserResult | A string including the user name, user account UID, and user description. |

**Usage Notes**

All of the non-required fields will be modified if a value is passed in that is different than what exists.

**PayValidationProviderInvoice**

This web method allows a specific invoice to be paid.

**Input XML Parameters**

| Parameter                                  | Description   | Required |
|--|---|----------|
| ExternalUserAccountName                    | 6 digit user name of the external validation provider user with the Pay Invoice privilege.                                | Yes      |
| ExternalUserAccountPassword                | Password of the external validation provider user.  | Yes      |
| InvoiceUid                                 | UID of the invoice to be paid.  | Yes      |
| PaymentMethodUid                           | UID of the payment method.  | Yes      |
| AccountCredits/<br>AccountCreditsListParam | Complex type* collection of the following parameters: <ul style="list-style-type: none"> <li>AccountCreditsUid</li> </ul> | No       |
| PaymentAmount                              | Amount of the payment.  | Yes      |
| CashDrawerUid                              | UID of the Web drawer.  | Yes      |
| ReferenceNote                              | Reference note for the payment.   | No       |

**Return XML Parameters**

| Parameter                          | Description  |
|------------------------------------|--|
| PayValidationProviderInvoiceResult | A string including the receipt UID, third party amount due, third party amount invoiced, and third party amount un-invoiced. |

**Usage Notes**

\*Complex Type – This type of parameter contains nested parameters, as indicated in the table.

## RemoveValidationProviderUser

Unlinks an external system user account from a validation provider. This web method will not delete the user account.

### Input XML Parameters

| Parameter                   | Description  | Required |
|-----------------------------|--|----------|
| ExternalUserAccountName     | 6 digit user name of the external validation provider user with the Validation Provider User Management privilege. | Yes      |
| ExternalUserAccountPassword | Password of the external validation provider user.   | Yes      |
| UserUid                     | UID of external user in Flex.  | Yes      |

### Return XML Parameters

| Parameter                          | Description               |
|------------------------------------|---------------------------|
| RemoveValidationProviderUserResult | User account UID and name |

### Usage Notes

The Validation Provider is determined via the External System User Account.

## T2\_FLEX\_VEHICLES

### InsertUpdateCustomFlds() v1.0

#### Input XML Parameters

| Parameter    | Description                    | Required |
|--------------|--------------------------------|----------|
| (FIELD_NAME) | Custom Field Name (Ex: Cust_1) | Yes      |
| VEHICLE_UID  | Vehicle UID (VEHICLE.VEH_UID)  | Yes      |

#### Return XML Parameters

| Parameter   | Description                                  |
|-------------|--|
| VEHICLE_UID | UID of vehicle that was successfully updated |

### Usage Notes

This method populates an existing custom field on the vehicle record specified by VEHICLE\_UID. The parameter name designated by {FIELD\_NAME} is the name of the custom field to be used. The parameter value is the value to be inserted.

More than one field can be passed by including multiple field names among the XML parameters.

**InsertUpdateVehicle() v1.0****Input XML Parameters**

| Parameter           | Description   | Required                             |
|---------------------|---|--------------------------------------|
| VEHICLE_UID         | Vehicle UID (VEHICLE.VEH_UID). This parameter is only sent if an existing vehicle is being updated. | No, if inserting<br>Yes, if updating |
| STATE               | Vehicle license plate state (STATE_MLKP.STL_UID). This is the UID for the state as found in Flex.   | Yes, if inserting<br>No, if updating |
| LICENSE_PLATE       | Vehicle license plate number. This parameter is ignored if an existing vehicle is being updated.    | Yes, if inserting<br>No, if updating |
| PLATE_TYPE          | Vehicle license plate type code (VEH_PLATE_TYPE_LKP.VPL_UID)  | Yes, if inserting<br>No, if updating |
| PLATE_REG_EXP_MONTH | Month of license plate registration expiration. Format: MM  | No                                   |
| PLATE_REG_EXP_YEAR  | Year of License plate registration expiration.<br>Format: YYYY                                      | No                                   |
| VIN                 | Vehicle VIN number  | No                                   |
| VEHICLE_MAKE        | Vehicle make code (VEH_MAKE_LKP.VKL_UID)  | No                                   |
| VEHICLE_MODEL       | Vehicle model code (VEH_MODEL_MLKP.VML_UID)   | No                                   |
| VEHICLE_STYLE       | Vehicle style code (VEH_STYLE_LKP.VSL_UID)  | No                                   |
| VEHICLE_COLOR       | Vehicle color code.(VEH_COLOR_LKP.VCL_UID)  | No                                   |
| VEHICLE_YEAR        | Vehicle Year of Manufacture. Format: YYYY   | No                                   |
| (FIELD_NAME)        | Custom Field Name   | No                                   |

**Return XML Parameters**

| Parameter   | Description  |
|-------------|--|
| VEHICLE_UID | UID of vehicle that was successfully inserted or updated |



### Usage Notes

This method is used to insert a new vehicle record or to modify an existing vehicle record.

If a vehicle UID is not passed, then the method will attempt to insert a new vehicle using the parameters passed into the method.

If a vehicle UID is passed, then method will attempt to update an existing vehicle with the parameters passed into the method. During an update the vehicle license plate will not be modified. If a value is passed in parameter LICENSE\_PLATE for an update, it will be ignored, and the existing plate number will be retained on the vehicle. (If a new license plate number has been assigned to an existing vehicle, use this method to insert a record for the new record for the vehicle rather than attempting to update the existing record; the two records may be merged manually by a T2 Flex user).

## InsertUpdateVehicleAssociation() v1.0

### Input XML Parameters

| Parameter             | Description  | Required                             |
|-----------------------|--|--------------------------------------|
| ASSOCIATION_UID       | Vehicle Association UID (ENT_VEH_REL.EVR_UID)  | No, if inserting<br>Yes, if updating |
| VEHICLE_UID           | Vehicle UID (VEHICLE.VEH_UID)  | Yes, if inserting<br>No, if updating |
| ENTITY_UID            | Customer UID (ENTITY.ENT_UID). This parameter is ignored if an existing vehicle association is being updated.  | Yes, if inserting<br>No, if updating |
| ASSOC_TYPE            | Association type code (VEH_ASSOCIATION_LKP.VAL_UID)  | Yes, if inserting<br>No, if updating |
| ASSOC_START_DATE      | Association start date. Format: MM/DD/YY   | Yes, if inserting<br>No, if updating |
| ASSOC_END_DATE        | Association end date. Format: MM/DD/YY   | No                                   |
| ASSIGN_CONTRAVENTIONS | Assign unassigned contraventions. Format: 1 for yes or 0 for no. If no value is sent then the citation will remain unassigned. If 1 is sent then the customer associated with this | No                                   |

|                |  |    |
|----------------|--|----|
|                | relationship will be made the responsible customer for all citations for this vehicle.   |    |
| ASSOC_PRIORITY | Association priority. Format is integer, with 1 representing highest priority relative to other associations for this vehicle. Default value is 1 for insert, or value of the existing priority for an update. | No |

### Return XML Parameters

| Parameter       | Description  |
|-----------------|--|
| ASSOCIATION_UID | UID of vehicle association that was successfully inserted or updated |

### Usage Notes

This method is used to insert a new vehicle association record or update an existing vehicle association record. A vehicle association is the record in Flex that defines the relationship between a vehicle and an entity.

If a vehicle association UID is not passed then the method will attempt to insert a new vehicle association using the information passed into the method.

If a vehicle association UID is passed then method will attempt to update an existing vehicle association using information passed into the method. If a value is passed in parameter ENTITY\_UID for an update it will be ignored. (If a new license plate number has been assigned to an existing vehicle, use this method to insert a record for the new record for the vehicle rather than attempting to update the existing record).

If an association priority is passed then the method will reprioritize all associations for the vehicle in order to place the association at the specified priority position; the association previously occupying that priority and all lower priority associations for the vehicle will have their priorities lowered. If a priority greater than that of any existing customer vehicle association is passed, then the association should be moved to the last position.

If no association priority is passed on an insert, the vehicle association will be inserted at priority position 1. If no customer vehicle association priority is passed on an update, the priority of the association will be unchanged.

## T2\_FLEX\_WAITLISTS

### GetEligibleWaitlists() v1.0

This method returns a list of all eligible waitlists for a given customer.

Note: This method can be extended to use customized business logic with a professional services engagement. In order to extend this method, a custom DLL must be built by a technical resource at T2.

**Input XML Parameters**

| Parameter | Description  | Required |
|-----------|--------------|----------|
| EntityUid | Customer UID | Yes      |

**Return XML Parameters**

| Parameter                  | Description   |
|----------------------------|---|
| GetEligibleWaitlistsResult | Indicates in the results string which waitlists the customer is eligible for. |

**InsertWaitlistRequest() v1.0****Input XML Parameters**

| Parameter         | Description                                  | Required  |
|-------------------|--|---|
| ENTITY_UID        | Customer UID (ENTITY.ENT_UID)                | Yes   |
| WAITLIST_UID      | Waitlist UID (WAITLIST.WLT_UID)              | Yes   |
| WAITLIST_PRIORITY | Waitlist Priority                            | No- If no value is submitted the new entry will be placed at the top of the list. |
| REQUEST_DATE      | Waitlist request date. Format is MM/DD/YYYY. | No- If no value is submitted the current time and date will be used.              |

**Return XML Parameters**

| Parameter            | Description           |
|----------------------|-----------------------|
| WAITLIST_REQUEST_UID | Waitlist Request UID. |

**Usage Notes**

This method is used to insert a new waitlist request. If a request date is not included the current date and time will be used. The priority parameter will observe the following logic:

| Value   | Action                                       |
|---|--|
| 1   | New request is placed at top of list.        |
| 9999  | New request is placed at bottom of list.     |
| Any number between first and last request position on list. | New request is placed at position requested. |
| Any number larger than last request position.               | New request is placed at bottom of list.     |

|      |                                       |
|------|---------------------------------------|
| Null | New request is placed at top of list. |
|------|---------------------------------------|

After a request is inserted all other request will be reprioritized and renumbered.

On the successful completion of this method an XML string will be returned containing the waitlist request UID. If the method cannot be completed successfully an error will be returned.

## ModifyWaitlistPriority() v1.0

### Input XML Parameters

| Parameter         | Description                     | Required  |
|-------------------|---------------------------------|---|
| ENTITY_UID        | Customer UID (ENTITY.ENT_UID)   | Yes   |
| WAITLIST_UID      | Waitlist UID (WAITLIST.WLT_UID) | Yes   |
| WAITLIST_PRIORITY | Waitlist Priority               | No- If no value is submitted the new entry will be placed at the top of the list. |

### Return XML Parameters

| Parameter    | Description   |
|--------------|---------------|
| WAITLIST_UID | Waitlist UID. |

### Usage Notes

This method is used to modify the priority of an existing waitlist request. The priority parameter will follow the following logic:

| Value   | Action                                   |
|---|--|
| 1   | Request is placed at top of list.        |
| 9999  | Request is placed at bottom of list.     |
| Any number between first and last request position on list. | Request is placed at position requested. |
| Any number larger than last request position.               | Request is placed at bottom of list.     |
| Null  | Request is placed at top of list.        |

After a request priority is changed all other requests will be reprioritized and renumbered.

## RemoveWaitlistRequest() v1.0

### Input XML Parameters

| Parameter            | Description                                | Required |
|----------------------|--|----------|
| ENTITY_UID           | Customer UID (ENTITY.ENT_UID)              | Yes      |
| WAITLIST_REQUEST_UID | Waitlist Request UID (WLT_ATTEMPT.WLA_UID) | Yes      |

### Return XML Parameters

| Parameter       | Description   |
|-----------------|---|
| REQUEST_REMOVED | This parameter with value = 1 Indicates that the specified waitlist request was successfully removed. |

### Usage Notes

This method is used to remove an existing waitlist request.

After a request is removed all other requests will be reprioritized and renumbered.

## T2\_FLEX\_WIRELESS

These methods are used to retrieve vehicle and permit information for real-time wireless communication on the handhelds. They are for internal T2 Systems use only and should only be used under the direction of Technical Support.

## CHAPTER 4: EXTERNAL WEB SERVICES

### EXTERNAL CUSTOM ACTION WEB SERVICES

T2 Flex allows you to configure custom actions that call an external system at certain points in your business processes. Currently, custom actions can only be configured for permits at the point of sale. The interface is based on a vendor-neutral Web Service model over which you have full control. However, your Web Methods **MUST** contain the following general inputs and outputs in order for T2 Flex to call and consume them appropriately.

#### Custom Action Web Method

An External Custom Action Web method must be self contained. All actions to be performed on custom fields must be performed by the external web method utilizing the existing T2\_FLEX\_Permission.InsertUpdateCustomFields web method provided by T2.

##### Input XML Parameters

The parameters for this Web Method can vary depending on the purpose of the web method. Each column specified in the Input Query (selected on the Custom Action Configuration lookup table in T2 Flex) will correspond to an XML node sent to the external Web Method.

##### Return XML Parameters

| Field             | Description  |
|-------------------|--|
| SUCCESS           | An integer value used to determine success or failure of the external action. Value = 1<br><br>Note: If the ERROR_NUMBER and ERROR_DESCRIPTION are returned, this field will not be returned (and vice versa). |
| ERROR_NUMBER      | The error number associated with the ERROR_DESCRIPTION. This indicates that the external Web Method has encountered a processing error or has broken internal business logic.                                  |
| ERROR_DESCRIPTION | The error description associated with the ERROR_NUMBER.<br><br>Note: If multiple ERROR_NUMBER fields are returned, multiple corresponding descriptions will be returned as well.                               |

##### Usage Notes

When invoking the Custom Action Web Method, T2 Flex will populate these fields with the values specified on the Custom Action Configuration lookup table:

- **XmlData:** Contains the necessary XML data to be consumed by the Web Method.
- **Version:** The version of the Web Method to use.
- **UserName:** The user name against which to authenticate.
- **Password:** The password against which to authenticate the given user.

T2 Flex will be performing an HTTP Post to call the external Web Methods. The external Web Service will need to be configured to allow HTTP Posts. For example, if the external Web Service is implemented in .NET, the following should be added to the Web Service's Web.Config file:

```
<configuration>
<system.web>
<webServices>
<protocols>
<add name="HttpPost"/>
</protocols>
</webServices>
</system.web>
</configuration>
```

## EXTERNAL PERMIT ELIGIBILITY WEB SERVICES

A customer's eligibility for Permit sales is determined using internal T2 Flex rules based on a customer's sub-classification, sales limits, as well as additional internal logic. To provide the maximum amount of flexibility, an external interface allows you to determine your own eligibility rules for tracked permits. The eligibility interface is based on a vendor-neutral Web Service model over which you have full control. However, your Web Methods **MUST** contain the following general inputs and outputs in order for T2 Flex to call and consume them appropriately.

### Number Range Eligibility Web Method

The Number Range Eligibility Web Method is used by T2 Flex to return a list of permit number ranges for which a customer is eligible. The parameters for this Web Method can vary depending on the permit number range's eligibility rules.

#### Input XML Parameters

| Parameter | Description   | Required?           |
|-----------|---|---------------------|
| ENT_UID   | The UID for the customer whose permit number range eligibility is being determined.               | Yes                 |
| PNA_UID   | The UID for the permit number range(s) for which the customer is eligible to purchase in T2 Flex. | No, only for permit |

|              |  |   |
|--------------|--|---|
|              |  | number ranges set to <b>T2 Flex Rules External Override</b> |
| Other fields | Additional T2 Flex fields from the Number Range Input Query (defined in Permit System Settings) may be provided. An XML field will be created for each field specified in the query. Each column in the query will correspond to a parameter sent to the Web Method. | No  |

### Return XML Parameters

| Parameter         | Description   |
|-------------------|---|
| PNA_UID           | The UID for the permit number range(s) for which the customer is eligible to purchase, provided in ascending order.<br><br>If ERROR_NUMBER and ERROR_DESCRIPTION are returned, this field will not be returned.   |
| SYSTEM_MSG        | A message indicating status, warning, or other details that will be displayed to the T2 Flex user.<br><br>This field will occur once and cannot be repeated.<br><br>If ERROR_NUMBER and ERROR_DESCRIPTION are returned, this field will not be returned.                              |
| ERROR_NUMBER      | If the Web Method encounters a processing error or has broken internal business logic, one or more error numbers will be returned along with corresponding error descriptions.<br><br>If PNA_UID or SYSTEM_MSG are returned, ERROR_NUMBER and ERROR_DESCRIPTION will not be returned. |
| ERROR_DESCRIPTION | The error description associated with the ERROR_NUMBER.<br><br>If PNA_UID or SYSTEM_MSG are returned, ERROR_NUMBER and ERROR_DESCRIPTION will not be returned.  |

### Usage Notes

When invoking the Number Range Eligibility Web Method, T2 Flex will populate these fields with the values specified in Permit System Settings:

- **XmlData:** Contains the necessary XML data to be consumed by the Web Method.



- **Version:** The version of the Web Method to use.
- **UserName:** The user name against which to authenticate.
- **Password:** The password against which to authenticate the given user.

T2 Flex will be performing an HTTP Post to call the external Web Methods. The external Web Service will need to be configured to allow HTTP Posts. For example, if the external Web Service is implemented in .NET, the following should be added to the Web Service's Web.Config file:

```
<configuration>
<system.web>
<webServices>
<protocols>
<add name="HttpPost"/>
</protocols>
</webServices>
</system.web>
</configuration>
```

## Single Permit Eligibility Web Method

The Single Permit Eligibility Web Method is used by T2 Flex to determine eligibility for the sale of a specific permit to a customer. The parameters for this Web Method can vary depending on the permit number range's eligibility rules.

### Input XML Parameters

| Parameter        | Description   | Required?  |
|------------------|---|--|
| PER_UID          | The UID of the permit whose eligibility is being determined.<br><br>For Non-Inventoried Permits, this field will be blank since the UID is not assigned until after the permit is sold. | Yes  |
| PEC_UID          | The UID of the Control Group whose permit eligibility is being determined.  | Yes  |
| ENT_UID          | The UID of the customer whose permit eligibility is being determined.   | Yes  |
| FLEX_ELIGIBILITY | Boolean value indicating whether or not the customer is eligible for the permit based on T2 Flex's business logic.<br><br>1 = eligible, 0 = not eligible                                | No, only for permit number ranges set to <b>T2 Flex Rules External</b> |

|                                 |  | <b>Override</b>   |
|---------------------------------|--|---|
| VIOLATES_SUBCLASS_ELIGIBILITY   | Boolean value indicating whether or not the customer violates T2 Flex's subclassification eligibility for the permit.<br><br>1 = violates, 0 = does not violate  | No, only for permit number ranges set to <b>T2 Flex Rules External Override</b> |
| VIOLATES_SALES_LIMITS           | Boolean value indicating whether or not the sale of the permit to the customer violates T2 Flex's subclassification sales limit for the permit.<br><br>1 = violates, 0 = does not violate  | No, only for permit number ranges set to <b>T2 Flex Rules External Override</b> |
| VIOLATES_SALES_CAPACITY         | Boolean value indicating whether or not the sale of the permit to the customer violates T2 Flex's sales capacity for the permit.<br><br>1 = violates, 0 = does not violate   | No, only for permit number ranges set to <b>T2 Flex Rules External Override</b> |
| VIOLATES_WAITLIST_RULES         | Boolean value indicating whether or not the sale of the permit to the customer violates T2 Flex's Wait List rules for the permit (i.e. whether or not they have been awarded the <b>right</b> to purchase the permit).<br><br>1 = violates, 0 = does not violate | No, only for permit number ranges set to <b>T2 Flex Rules External Override</b> |
| VIOLATES_PROPERTY_RULES         | Boolean value indicating whether or not the sale of the permit to the customer violates T2 Flex's Property rules for the permit.<br><br>1 = violates, 0 = does not violate   | No, only for permit number ranges set to <b>T2 Flex Rules External Override</b> |
| VIOLATES_PERMIT_ALLOTMENT_RULES | Boolean value indicating whether or not the sale of the permit to the customer violates T2 Flex's Permit Allotment rules for the permit.<br><br>1 = violates, 0 = does not violate   | No, only for permit number ranges set to <b>T2 Flex Rules</b>                   |

|              |   | <b>External Override</b> |
|--------------|---|--------------------------|
| Other fields | Additional T2 Flex fields from the Single Permit Eligibility Input Query (defined in Permit System Settings) may be provided. An XML field will be created for each field specified in the query. Each column in the query will correspond to a parameter sent to the Web Method. | No                       |

**Return XML Parameters**

| <b>Parameter</b>            | <b>Description</b>  |
|-----------------------------|---|
| IS_ELIGIBLE                 | Boolean value indicating whether or not the permit has been deemed eligible by the Single Permit Eligibility Web Method.<br><br>1 = eligible, 0 = not eligible  |
| NON_ELIGIBLE_REASON         | Text value describing why the customer was not eligible for the given permit.<br><br>This parameter is only returned (and is required) if IS_ELIGIBLE is set to 0.  |
| NOT_ELIGIBLE_SUBCLASS       | Boolean value indicating T2 Flex should treat the given permit as if it had violated T2 Flex subclassification eligibility.<br><br>1 = violates subclassification eligibility, 0 = does not violate<br><br>This parameter is only returned (and is required) if IS_ELIGIBLE is set to 0. If this value is 1, it will override any value in the NON_ELIGIBLE_REASON parameter. |
| NOT_ELIGIBLE_SALES_LIMIT    | Boolean value indicating T2 Flex should treat the given permit as if it had violated T2 Flex subclassification sales limit.<br><br>1 = exceeds limit, 0 = within limit<br><br>This parameter is only returned (and is required) if IS_ELIGIBLE is set to 0. If this value is 1, it will override any value in the NON_ELIGIBLE_REASON parameter.                              |
| NOT_ELIGIBLE_SALES_CAPACITY | Boolean value indicating T2 Flex should treat the given permit as if it had exceeded T2 Flex sales capacity.<br><br>1 = exceeds capacity, 0 = within capacity<br><br>This parameter is only returned (and is required) if IS_ELIGIBLE is set to 0. If this value is 1, it will override any value in the NON_ELIGIBLE_REASON  |

|   |   |
|---|---|
|   | parameter.  |
| NOT_ELIGIBLE_WAITLIST_RULES                   | <p>Boolean value indicating T2 Flex should treat the given permit as if it had violated T2 Flex Wait List rules.</p> <p>1 = violates, 0 = does not violate</p> <p>This parameter is only returned (and is required) if IS_ELIGIBLE is set to 0. If this value is 1, it will override any value in the NON_ELIGIBLE_REASON parameter.</p>        |
| NOT_ELIGIBLE_PROPERTY_RULES                   | <p>Boolean value indicating T2 Flex should treat the given permit as if it had violated T2 Flex Property rules.</p> <p>1 = violates, 0 = does not violate</p> <p>This parameter is only returned (and is required) if IS_ELIGIBLE is set to 0. If this value is 1, it will override any value in the NON_ELIGIBLE_REASON parameter.</p>         |
| NOT_ELIGIBLE_PERMIS-<br>SION_ALLOCATION_RULES | <p>Boolean value indicating T2 Flex should treat the given permit as if it had violated T2 Flex Permit Allotment rules.</p> <p>1 = violates, 0 = does not violate</p> <p>This parameter is only returned (and is required) if IS_ELIGIBLE is set to 0. If this value is 1, it will override any value in the NON_ELIGIBLE_REASON parameter.</p> |

### Usage Notes

When invoking the Single Permit Eligibility Web Method, T2 Flex will populate these fields with the values specified in Permit System Settings:

- **XmlData:** Contains the necessary XML data to be consumed by the Web Method.
- **Version:** The version of the Web Method to use.
- **UserName:** The user name against which to authenticate.
- **Password:** The password against which to authenticate the given user.

T2 Flex will be performing an HTTP Post to call the external Web Methods. The external Web Service will need to be configured to allow HTTP Posts. For example, if the external Web Service is implemented in .NET, the following should be added to the Web Service's Web.Config file:

```
<configuration>
<system.web>
<webServices>
<protocols>
```

```

<add name="HttpPost"/>
</protocols>
</webServices>
</system.web>
</configuration>

```

## EXTERNAL SALES WEB SERVICES

External sales interface functionality enables T2 Flex to act as the POS station for any external system and maintain a record of the sale within a customer record in T2 Flex. This feature makes use of a single Web Service defined in **System Settings > Interfaces** that contains the following two Web Methods. The interface is based on a vendor neutral Web Services model over which you have full control. However, your Web Methods **MUST** contain the following inputs and outputs in order for T2 Flex to call and consume them appropriately.

### Record/Adjust External Sale Web Method

The Record/Adjust External Sale Web Method is used by T2 Flex to allow external systems to insert sales items or make adjustments on existing items.

#### Input XML Parameters

| Parameter Name         | Description   | Required?                        |
|------------------------|---|----------------------------------|
| EXT_SALE_ITEM_TYPE_UID | A UID that matches the type of external sales item being sold.  | Yes, if EXT_TRANS_TYPE is 1      |
| EXT_SALE_ITEM_UID      | A UID matching an existing external sale item.  | Yes                              |
| EXT_TRANS_TYPE         | 1 = New sale (default if the field is omitted)<br>2 = Adjustment<br>3 = Alter Custom Field(s)   | No                               |
| AMOUNT                 | If EXT_SALE_TYPE is 1, then this is the initial amount of the external item. This value can be negative if the sale item type allows negative values. If the EXT_SALE_TYPE is 2, the amount determines the amount used to adjust the sale item and can be | Yes, if EXT_TRANS_TYPE is 1 or 2 |

|                  |   |   |
|------------------|---|---|
|                  | negative as well.   |   |
| ADJ_REASON       | If EXT_TRANS_TYPE = 2, this field specifies the reason for the adjustment. The reason is added to the financial history for the external sale item. | No                                      |
| ENT_UID          | The UID of the entity who should be assigned the external sales item.   | No                                      |
| ENT_PRIMARY_ID   | The primary ID of the entity who should be assigned the external sales item.  | No                                      |
| ENT_SECONDARY_ID | The secondary ID of the entity who should be assigned the external sales item.  | No                                      |
| ENT_TERTIARY_ID  | The tertiary ID of the entity who should be assigned the external sales item.   | No                                      |
| Custom Fields    | Custom fields can be passed into the web method and are identified by their field names in T2 Flex.   | Determined by Data Dictionary Settings. |

### Return XML Parameters

| Parameter Name    | Description   |
|-------------------|---|
| EXT_SALE_ITEM_UID | The UID of the external sales item created by T2 Flex.<br>If ErrorNumber and ErrorDescription are returned, this field will not be returned (and vice versa).   |
| ErrorNumber       | A code indicating an error occurred during the insertion or adjustment process.   |
| ErrorDescription  | A description of the error that occurred during the insertion or adjustment process.<br>If multiple ErrorNumber fields are returned, multiple corresponding descriptions will be returned as well.<br>If EXT_SALE_ITEM_UID is returned, no errors will be |

|  |           |
|--|-----------|
|  | returned. |
|--|-----------|

### Usage Notes

When invoking the Record/Adjust External Sale Web Method, T2 Flex will populate these fields with the values specified in Interfaces System Settings:

- **XmlData:** Contains the necessary XML data to be consumed by the Web Method.
- **Version:** The version of the Web Method to use.
- **UserName:** The user name against which to authenticate.
- **Password:** The password against which to authenticate the given user.

Although none of the customer related fields (ENT\_UID, ENT\_PRIMARY\_ID, ENT\_SECONDARY\_ID and ENT\_TERTIARY\_ID) are required, *T2 Flex must match exactly one customer when trying to record an external sale.* It will search for the customer using all supplied fields, in order. If no match is found, or multiple matches are found, an error is returned and the web method will fail.

Zero balance external items cannot be adjusted. An external sales item also cannot be “adjusted past zero”. If an external sales item has an initial negative balance, it cannot be adjusted higher than zero. Similarly, if an external sales item has an initial positive balance; it cannot be adjusted lower than zero.

Custom fields can be set when the sale item is inserted and can also be altered using a subsequent call to the Web Method if desired. Flex compares each custom field specified in the XML input with the current value of the custom field. If there is a difference, Flex will alter the custom field to be the new value.

### Possible Errors Returned

- EXT\_SALE\_ITEM\_TYPE\_UID is missing or invalid.
- EXT\_TRANS\_TYPE is missing or invalid.
- AMOUNT is missing or invalid.
- Cannot find entity (to attach the external sales item).
- Found more than one Entity.
- Cannot find external sales item (for an adjustment).
- Data dictionary error (if a field doesn’t match a data dictionary rule, such as a regular expression or uniqueness constraint).

## Items Sold Web Method

The Items Sold Web Method is used by T2 Flex to reconcile sales from T2 Flex to the external system. It informs the external system that payment has been received for items purchased by a customer. It can also be used when a receipt is reversed that includes external sale item(s) by sending a negative value for the AMOUNT\_PAID.

**Input XML Parameters**

| Parameter Name         | Description  | Required ?                              |
|------------------------|--|---|
| EXT_SALE_ITEM_TYPE_UID | The UID of the type of item being sold.  | Yes                                     |
| EXT_SALE_ITEM_UID      | The UID of the external sale item that was sold.   | Yes                                     |
| ERQ_UID                | The UID of the record in the EXT_SALE_ITEM_RESEND_REL table.   | Yes                                     |
| DESCRIPTION            | The description of the item.   | No                                      |
| DATE_PAID              | The date on which the item was paid.   | No                                      |
| AMOUNT_PAID            | The amount that was paid for the item, excluding taxes.  | No                                      |
| TAX_1_AMOUNT           | The amount that was paid for the item that was taxed by tax 1.   | No                                      |
| TAX_2_AMOUNT           | The amount that was paid for the item that was taxed by tax 2.   | No                                      |
| TAX_3_AMOUNT           | The amount that was paid for the item that was taxed by tax 3.   | No                                      |
| TRANSACTION_COMPLETE   | A Boolean flag indicating if T2 Flex considers the item paid for in full.  | No                                      |
| REC_UID                | The UID of the receipt for the item paid.  | No                                      |
| Custom fields          | All custom fields associated with the external sales item will be sent as input parameters to this web method.   | Determined by Data Dictionary Settings. |
| Other fields           | Other fields from the input query specified in Interfaces System Settings can appear in the input parameter list. The field names in the XML correspond to the field names in the query.<br><br>For example,<br>REC_PAYMENT_METHOD_DISPL |   |



|  |   |  |
|--|---|--|
|  | AY might be passed in to display the payment method used in plain text. |  |
|--|---|--|

### Return XML Parameters

| Parameter Name    | Description   |
|-------------------|---|
| EXT_SALE_ITEM_UID | The External Sale Item UID for the sale. Since one receipt can hold more than financial transaction, the External Sale Item UID is required to identify which transaction in the receipt was processed. |
| ERQ_UID           | The UID of the Record from the EXT_SALE_ITEM_RESEND_REL table. This is required for T2 Flex to match the transaction back to its own records.   |
| RESPONSE_CODE     | 1 = Indicates the External Sales Item is accepted.<br>2 = Indicates the External Sales Item should be re-sent later.  |
| ErrorNumber       | A code indicating an error occurred during the insertion or adjustment process.   |
| ErrorDescription  | A description of the error that occurred during the insertion or adjustment process.  |

### Usage Notes

When invoking the Items Sole Web Method, T2 Flex will populate these fields with the values specified in Interfaces System Settings:

- **XmlData:** Contains the necessary XML data to be consumed by the Web Method.
- **Version:** The version of the Web Method to use.
- **UserName:** The user name against which to authenticate.
- **Password:** The password against which to authenticate the given user.

The web method should return one record for each item submitted, encapsulated inside of an <Item> node, indicating the status of the item.

### Possible Errors Returned

The errors returns from this web method depend on the implementation of the external system.

## CHAPTER 5: WEB SERVICES ERROR MESSAGES

These error messages are generated by Web services, and are not the only possible errors that may be returned.

Additional errors, generated by the T2 Flex Business Object Layer or by the Oracle database layer may also be returned and are not catalogued in this document. Developers using Web services should trap for any error, not just the specific error numbers listed below.

### Common Error Messages

| Code   | Description  |
|--|--|
| 9000, 9007, 9021, 9023, 9100, 9200, 9205, 9300, 9310, 9330, 9353, 9400, 9420, 9440, 9460, 9480, 9500, 9540, 9560, 9590, 9620, 9650, 9670, 9690, 9700, 9720, 9751, 17011, 17081, 17200, 17401             | Unsupported version provided.  |
| 5452, 9001, 9008, 9022, 9024, 9101, 9201, 9206, 9301, 9311, 9331, 9354, 9401, 9421, 9441, 9461, 9481, 9501, 9541, 9561, 9591, 9621, 9651, 9671, 9691, 9701, 9720, 9725, 9752, 17012, 17082, 17201, 17402 | User authentication failed.  |
| 6000, 9034, 9035, 9036, 9105, 9207, 9208, 9313, 9343, 9710   | The authenticated user does not have the privilege to perform this action. |
| 9041, 9106, 9215, 9317, 9371, 9722, 9755, 17006, 17080, 17206, 17400   | General web method error – XXX.  |
| 9045, 9102, 9214, 9316, 9355, 9402, 9422, 9442, 9462, 9482, 9502, 9542, 9562, 9592, 9622, 9651, 9672, 9692, 9753, 17000, 17000   | XmlData format error.  |
| 9006, 9020, 9033, 9039, 9204, 9342, 9709   | Data type mismatch.  |
| 9115, 9120, 9545, 17403  | Custom Field Error: XXX.   |
| 9103, 9361, 9362, 9449, 9537, 9636, 9656, 9676, 9697, 9754   | T2 Flex Error- XXX.  |

| Code   | Description                                    |
|--|--|
| 9038, 9043, 9046, 9216, 9318, 9533, 9723, 9764, 9784 | XXX is an invalid value for the YYY parameter. |

### Specific Error Messages

| Code | Description  |
|------|--|
| 1111 | Must have a Misc Item UID to purchase.   |
| 1201 | The appeal is not in the correct workflow stage to be scheduled.   |
| 1406 | The citation is not in the correct status to be reduced to a warning (if applying appeal results will reduce the citation to a warning). |
| 1408 | The citation is not in the correct status to be voided (if applying appeal results will void the citation).                              |
| 1501 | Date of docket has passed.   |
| 1511 | An officer is required on the docket and is not the officer on the primary citation or boot/tow linked to the given appeal.              |
| 1518 | Docket lead time has not been met.   |
| 1902 | An error occurred (Miscellaneous Item UID) has an invalid category [NUMber] and cannot be added to the basket.)                          |
| 1909 | The [MISC ITEM CODE] requires an override.   |
| 2218 | <Permit Number> cannot be returned.  |
| 2266 | The refund amount must be greater or equal to zero.  |
| 2335 | Invalid return date.   |
| 4303 | Payment Method ( ) is not valid for this basket.   |
| 4401 | The Amount Per Payment value is invalid.   |
| 4509 | The payment method is not active.  |
| 9002 | Type is required for a Customer insert.  |
| 9003 | SubClassification is required for a Customer insert.   |
| 9004 | Type is required for a Customer and cannot be cleared.   |
| 9005 | SubClassification is required for a Customer and cannot be cleared.  |
| 9009 | Customer is required for an Address insert or update.  |
| 9010 | Type is required for an Address insert.  |
| 9011 | PrimaryStreet is required for an Address insert.   |
| 9014 | StartDate is required for an Address insert.   |
| 9015 | Type is required for an Address and cannot be cleared.   |
| 9016 | PrimaryStreet is required for an Address and cannot be cleared.  |
| 9019 | StartDate is required for an Address and cannot be cleared.  |

| Code  | Description   |
|---|---|
| 9040  | Customer UID XXX is not related to Address UID XXX.                                   |
| 9025  | Customer is required for an Email Address insert.                                     |
| 9026  | Type is required for an Email Address insert.   |
| 9027  | Email is required for an Email Address insert.  |
| 9028  | StartDate is required for an Email Address insert.                                    |
| 9029  | Type is required for an Email Address and cannot be cleared.                          |
| 9030  | Email is required for an Email Address and cannot be cleared.                         |
| 9031  | StartDate is required for an Email Address and cannot be cleared.                     |
| 9032  | RelatedEntity is required for an Email Address and cannot be cleared.                 |
| 9037  | The authenticated user does not have the privilege to perform this action.            |
| 9044  | Customer UID XXX is not related to Email Address UID XXX.                             |
| 9039  | The length of the data for field "XXX" is greater than the max length for that field. |
| 9042, 9408, 9426, 9491, 9511, 9565, 9595, 9760, 9770, 9780, 23000 | Customer UID XXX is invalid or missing.   |
| 9058  | AccountCredit Uid + this.AccountCreditsUID + " is not valid."                         |
| 9110  | State is required for a Vehicle insert.   |
| 9111  | LicensePlate is required for a Vehicle insert.  |
| 9114  | License Plate cannot be updated to a new value.                                       |
| 9116  | License Plate Type is required for a Vehicle insert.                                  |
| 9130  | AssociationType is required for a Vehicle Association insert.                         |
| 9131  | StartDate is required for a Vehicle Association insert.                               |
| 9132  | AssociationType is required for a Vehicle Association and cannot be cleared.          |
| 9133  | StartDate is required for a Vehicle Association and cannot be cleared.                |
| 9134  | Vehicle UID is required for a Vehicle Association insert.                             |
| 9135  | Customer UID is required for a Vehicle Association insert.                            |
| 9202  | AdjudicationType is required for an Adjudication insert.                              |
| 9203  | AdjudicationDate is required for an Adjudication insert.                              |
| 9209  | An Adjudication insert requires a RelatedEntity.                                      |
| 9212  | Citation number does not exist in database.   |
| 9213  | Adjudication Note Code UID XXX is invalid.  |

| Code       | Description  |
|------------|--|
| 9302       | ContraventionNumber is required for a Citation payment.  |
| 9303       | PaymentType is required for a Citation payment.  |
| 9304       | PaymentAmount is required for a Citation payment.  |
| 9305       | CashDrawer_UID is required for a Citation payment.   |
| 9306, 9337 | An error relating to the cash drawer has occurred.   |
| 9307, 9338 | The cash drawer required for this transaction could not be found.  |
| 9308       | Citation number @ could not be found.<br>Misc Item + this.MiscellaneousItemUID + " could not be found.",   |
| 9309       | Data type mismatch. Error description: +ex.Message   |
| 9312       | The authenticated user does not have the privilege to perform this action.                                 |
| 9314       | Amount paid must equal total amount due.   |
| 9315       | Citation UID is required.  |
| 9318       | this.PaymentMethodLookupUID.ToString() + " is an invalid value for the 'PaymentMethodLookupUID' parameter. |
| 9319       | Payment Plan Transfer Agency UID XXX is invalid or missing.  |
| 9320       | Payment Plan Time Period is invalid or missing.  |
| 9321, 9363 | Payment Plan Tax Type is invalid or missing.   |
| 9322, 9364 | Both the Number of Payments and Amount per Payment cannot be included in the XML string.                   |
| 9323, 9365 | Payment Plan Number of Payments is invalid or missing.   |
| 9324, 9366 | Payment Plan Template UID XXX is invalid.  |
| 9325, 9367 | Payment Plan Amount per Payment is invalid or missing.   |
| 9326, 9368 | Payment Plan Start Date is missing or invalid.   |
| 9327, 9369 | Either a Number of Payments or an Amount per Payment is required for Payment Plans.                        |
| 9328       | Payment Plan Customer UID is invalid or missing.   |
| 9329       | Payment Type UID cannot be used for web service citation sales.  |
| 9332       | ContraventionNumber is required for a Pre-Paid Citation payment.   |
| 9333       | Violation_UID is required for a Pre-Paid Citation payment.   |
| 9334       | PaymentType is required for a Pre-Paid Citation payment.   |
| 9335       | PaymentAmount is required for a Pre-Paid Citation payment.   |
| 9336       | CashDrawer_UID is required for a Pre-Paid Citation payment.  |
| 9339       | ContraventionNumber already exists and cannot be pre-paid.   |
| 9340       | Violation_UID is XXX invalid or missing.   |

| Code  | Description  |
|---|--|
| 9341  | State_UID XXX is invalid or missing.   |
| 9345  | Payment amount is more than the account balance available for use.   |
| 9347  | An Account Credit with a \$0.00 remaining balance cannot be used.  |
| 9348  | AccountCredit Uid + acp.AccountCreditsUID.ToString() + " is not a valid AccountCredit for Entity Uid " + this.EntityUID.ToString() |
| 9350  | Permit UID XXX is a Bulk Permit and cannot be modified using Web Services.   |
| 9351  | No valid Permit is available for sale in the specified Number Series.  |
| 9352  | Permit mailing tracking number value exceeds maximum field length.   |
| 9356, 9423, 9443, 9463, 9483, 9503, 9543, 9563, 9593, 9623, 9653, 9673, 9693              | Permit UID is required.  |
| 9357, 9406, 9424, 9444, 9464, 9484, 9504, 9544, 9564, 9594, 9624, 9654, 9674, 9694, 23000 | Permit UID XXX is invalid.   |
| 9358  | Receipt UID is invalid or missing.   |
| 9359  | Permit UID XXX was not sold on this receipt.   |
| 9360  | Permit UID XXX has already been confirmed.   |
| 9373  | Permit sale price XXX is invalid.  |
| 9374  | A payment type of "No Charge" must be used with a permit sale price of 0.00.   |
| 9375  | Permit effective date is required for control groups with a "Set at Point of Sale" effective date.                                 |
| 9376  | Permit expiration date is required for control groups with a "Set at Point of Sale" effective date.                                |
| 9377  | The Control Area parameter has been deprecated and should not be passed in to this method.   |
| 9378  | Control Group UID XXX requires at least one Facility, but none were provided.  |
| 9379, 9609  | Facility UID XXX is invalid.   |
| 9380  | Stall Type UID XXX cannot be provided without an Area UID.   |
| 9381  | Stall UID XXX cannot be provided without an Area UID.  |
| 9382  | Stall UID XXX cannot be provided without a Stall Type UID.   |
| 9383  | Stall Type UID XXX requires a Stall UID.   |

| Code                                     | Description  |
|--|--|
| 9389                                     | Allotment Facility UID X is invalid.   |
| 9390                                     | Allotment Facility UID is required.  |
| 9391                                     | Allotment Facility Group UID X is invalid.   |
| 9392                                     | Allotment Facility Group UID is required.  |
| 9393                                     | Facility UID X is not active and cannot be used for this transaction.                                      |
| 9394                                     | Facility Group UID X is not active and cannot be used for this transaction.                                |
| 9395                                     | Facility Group UID X is not valid for this transaction as there are no active facilities within the group. |
| 9396                                     | A valid permit allotment could not be identified for use with this permit sale.                            |
| 9398                                     | Payment Plan Transfer Agency UID + this.TransferAgencyLookupUID.ToString() + " is invalid or missing.      |
| 9403                                     | Permit Number Range UID XXX is invalid.  |
| 9404                                     | Permit Control Group UID XXX is invalid.   |
| 9405, 9488, 9508, 9628                   | Cash Drawer UID XXX is invalid or inactive.  |
| 9407                                     | Permit UID and Permit Number Range UID cannot both be included in the XmlData string                       |
| 9409                                     | Date value XXX is invalid.   |
| 9410                                     | Permit UID or Permit Number Range UID are either invalid or missing.                                       |
| 9411                                     | Control Group UID XXX has not completed setup.   |
| 9412, 9492, 9512, 9596, 9761, 9771, 9781 | Customer UID is required.  |
| 9413                                     | Permit UID XXX does not have a status of Allocated.  |
| 9414, 9445, 9514, 9601                   | Control Group UID is required.   |
| 9415, 9489, 9509, 9629                   | Cash Drawer UID is required.   |
| 9416                                     | Permit Number Range UID XXX has not completed setup.   |
| 9417                                     | Reservation End Date is required.  |
| 9418                                     | Cash Drawer UID XXX does not have any permits allocated from the Permit Number Range UID XXX.              |
| 9419, 9448                               | Control Group UID XXX is not related to Permit Number Range UID XXX and cannot be used.                    |
| 9425                                     | Permit UID XXX is a Bulk Permit and cannot be modified using Web Services.                                 |

| Code             | Description   |
|------------------|---|
| 9427             | Customer UID is required when ANONYMOUS USER ID is empty.   |
| 9446, 9513, 9600 | Control Group UID XXX is invalid.   |
| 9447, 9507, 9597 | Permit UID XXX does not have a status of Reserved.  |
| 9465             | Permit Fee UID is required.   |
| 9466             | Permit Fee UID XXX is invalid   |
| 9467             | Permit Fee XXX cannot be applied to permissions.  |
| 9468             | Permit UID XXX does not have a status of Active.  |
| 9469             | Permit Fee UID XXX does not have a status of Active.  |
| 9485, 9505, 9625 | Payment Type UID is required.   |
| 9486, 9506, 9626 | Payment Type UID XXX is invalid.  |
| 9487             | Permit UID XXX does not have a status of Active.  |
| 9490, 9510, 9630 | Payment amount is required.   |
| 9493             | Customer UID XXX does not current have a balance.   |
| 9494             | Customer UID XXX is not related to Permit UID XXX.  |
| 9495             | Payment amount of XXX does not equal permit balance of XXX.   |
| 9496, 9532       | Payment amount XXX is invalid.  |
| 9515             | Space Type UID XXX is invalid.  |
| 9517             | Linked Vehicle UID XXX is invalid.  |
| 9518             | Linked Property UID XXX is invalid.   |
| 9519             | Permit effective date format is invalid.  |
| 9520             | Permit effective date is required for control groups with a “Set at Point of Sale” effective date.  |
| 9521             | Permit expiration date format is invalid.   |
| 9522             | Permit expiration date is required for control groups with a “Set at Point of Sale” effective date. |
| 9523, 9598       | Permit UID XXX has not been reserved for Customer UID XXX.  |
| 9524, 9599       | Permit UID XXX reservation has expired.   |
| 9525, 9602       | Control Group UID XXX is not valid for Permit UID XXX.  |
| 9526, 9603       | Customer Subclassification UID XXX is not valid for Control Group UID XXX.                          |
| 9527, 9604       | Printable permits are unavailable for sale through Web services.                                    |
| 9528, 9605       | Non-tracked permits are unavailable for sale through Web services.                                  |
| 9529, 9606       | Control Group UID XXX has not completed setup.  |
| 9530             | Payment Type XXX cannot be used for Web service permit sales.                                       |
| 9531             | Payment Amount XXX does not equal the permit sale price of XXX.                                     |



| Code             | Description   |
|------------------|---|
| 9534             | Permit UID XXX is a Bulk Permit and cannot be modified using Web Services.                  |
| 9535             | Shipping Method UID is invalid or missing.  |
| 9536             | Mailing Address UID is invalid or missing.  |
| 9538             | Payment Plan Transfer Agency UID XXX is invalid or missing.                                 |
| 9539             | Payment Plan Time Period is invalid or missing.   |
| 9546             | The length of the data for field “XXX” is greater than the max length for that field.       |
| 9566             | Vehicle UID XXX is invalid.   |
| 9567             | Property UID XXX is invalid.  |
| 9568             | Customer Relationship UID XXX is invalid.   |
| 9569             | Vehicle Relationship UID XXX is invalid.  |
| 9570             | Property Relationship UID XXX is invalid.   |
| 9571             | An Object UID and an Object Relationship UID cannot both be included in the XmlData string. |
| 9572             | Relationship start date format is invalid.  |
| 9573             | Relationship end date format is invalid.  |
| 9574             | Relationship start date is required.  |
| 9575             | An Object UID or an Object Relationship UID is required.                                    |
| 9576             | An unspecified error has occurred and the Permit Relationship UID could not be created.     |
| 9577             | Permit UID XXX does not have a permit status that allows permit relationships.              |
| 9578             | Customer Relationship UID XXX is invalid for Permit UID XXX.                                |
| 9579             | Vehicle Relationship UID XXX is invalid for Permit UID XXX.                                 |
| 9580             | Property Relationship UID XXX is invalid for Permit UID XXX.                                |
| 9581             | A valid relationship between (Object Name) UID XXX and Permit UID XXX currently exists.     |
| 9607, 9663, 9683 | Permit UID XXX is a Bulk Permit and cannot be modified using Web Services.                  |
| 9608             | Control Group UID XXX requires at least one Facility, but none were provided.               |
| 9610             | Stall Type UID XXX cannot be provided without an Area UID.                                  |
| 9611             | Stall UID XXX cannot be provided without an Area UID.                                       |
| 9612             | Stall UID XXX cannot be provided without a Stall Type UID.                                  |
| 9613             | Stall Type UID XXX requires a Stall UID.  |

| Code       | Description  |
|------------|--|
| 9627       | Permit UID XXX does not have a status of Active.   |
| 9631       | Renewal Period UID is required.  |
| 9632       | Renewal Period UID XXX is invalid  |
| 9633       | Permit Effective Date format is invalid.   |
| 9634       | Permit Effective Date is required.   |
| 9635       | Permit UID XXX does not have a permit status that allows permit renewals.                  |
| 9637       | Payment Amount XXX is invalid.   |
| 9638       | Permit UID XXX is a Bulk Permit and cannot be modified using Web Services                  |
| 9655       | Missing Reason UID XXX is invalid  |
| 9657, 9677 | Address UID XXX is invalid.  |
| 9658, 9678 | Email Address UID XXX is invalid   |
| 9659, 9679 | Generate Letter value XXX is invalid.  |
| 9660, 9680 | When the value of Generate Letter is true then an Address UID or an Email UID is required. |
| 9661, 9681 | The Address UID XXX is not related to any of the customers on Permit UID XXX.              |
| 9662, 9682 | The Email Address UID XXX is not related to any of the customers on Permit UID XXX.        |
| 9675       | Deactivate Reason UID XXX is invalid   |
| 9695       | Permit mail date format is invalid.  |
| 9696       | Permit mail date is required.  |
| 9698       | Permit UID XXX is not a mailed permission.   |
| 9699       | Permit UID XXX has already been mailed.  |
| 9702       | NoteTypeUID is required for a Note insert.   |
| 9703       | Description is required for a Note insert.   |
| 9704       | ObjectType is required for a Note insert.  |
| 9705       | ObjectReferenceUID is required for a Note insert.  |
| 9706       | NoteTypeUID is required for a Note and cannot be cleared.                                  |
| 9707       | ObjectType is required for a Note and cannot be cleared.                                   |
| 9708       | ObjectReferenceUID is required for a Note and cannot be cleared.                           |
| 9727       | Query is not available for Web Service use.  |
| 9724       | Note text cannot exceed 2000 characters.   |
| 9762, 9782 | Waitlist UID XXX is invalid.   |

| Code       | Description   |
|------------|---|
| 9763, 9783 | Waitlist UID is required.   |
| 9765       | Waitlist request date format is invalid.  |
| 9772       | Waitlist Request UID XXX is invalid.  |
| 9773       | Waitlist Request UID is required.   |
| 9774       | The status of waitlist request UID XXX is set to “Expired” or “Is Right to Purchase” and cannot be removed. |
| 9775       | Customer UID XXX is not related to waitlist request UID XXX.  |
| 9785       | Waitlist UID XXX is not a current waitlist request for customer UID XXX.                                    |
| 9790       | Eligible waitlists general error.   |
| 9901       | The cash drawer for the appeal fees payment must be a WS drawer.  |
| 17007      | Customer cannot be added to allotment.  |
| 17008      | Customer is already linked to allotment.  |
| 17009      | Customer is not linked to allotment.  |
| 17010      | Customer cannot be unlinked from allotment.   |
| 17051      | Anonymous User ID XXX is invalid.   |
| 17052      | Permit Number Range UID is required.  |
| 17053      | Permit Control Group UID is required.   |
| 17054      | Quantity XXX is invalid.  |
| 17055      | Quantity is required.   |
| 17056      | XXX List does not align with the Permit UID list.   |
| 17057      | The Permit UID XXX is no longer reserved for the specified Anonymous User ID: YYY.                          |
| 17058      | Permit UID XXX is not reserved for the specified Anonymous User ID: YYY.                                    |
| 17059      | Relative Reservation Length XXX is invalid.   |
| 17060      | Event Title is missing.   |
| 17061      | Event Classification missing.   |
| 17062      | Event Start Date missing.   |
| 17063      | Event Start Time missing.   |
| 17064      | Event End Date missing.   |
| 17065      | Event End Time missing.   |
| 17066      | Event Title is too long.  |

| Code         | Description   |
|--------------|---|
| 17067        | Event Classification is invalid.  |
| 17068        | Start Date is invalid.  |
| 17069        | Start Time is invalid.  |
| 17070        | End Date is invalid.  |
| 17071        | End Time is invalid.  |
| 17072        | Event Description is too long.  |
| 17073        | Estimated Attendance is invalid.  |
| 17074        | Actual Attendance is invalid.   |
| 17075        | Event Discount is invalid.  |
| 17076        | Apply Tax1 is invalid.  |
| 17077        | Apply Tax2 is invalid.  |
| 17078        | Apply Tax3 is invalid.  |
| 17079        | Citation Location is invalid.   |
| 17087        | PPQ_UID_PRESOLD_PEC_REL is required.  |
| 17086        | PERMISSION_CTL_GRP_LIST and<br>PPQ_UID_PRESOLD_PEC_REL_LIST list size must be equal.                    |
| 17085        | Invalid Event Facility-Presold Relationship.  |
| 17101, 17111 | Facility Tiers cannot be validated with version XXX.  |
| 17102, 17112 | The specified control group XXX does not require facility tier selections.                              |
| 17103, 17113 | Non-Inventoried Permissions cannot be authorized/sold with version XXX.                                 |
| 17104, 17114 | The Permit cannot be authorized/sold through web services.  |
| 17105, 17115 | A Permission UID cannot be passed as a parameter when authorizing/selling a Non-Inventoried permission. |
| 17106, 17116 | The Non-Inventoried permit cannot be authorized/sold through web services when printing is enabled.     |
| 17107, 17117 | Invalid Facility Tier XXX.  |

| Code         | Description  |
|--------------|--|
| 17108, 17118 | The Facility XXX has been selected twice or more under different facility tiers.                 |
| 17109, 17119 | The minimum number of allowed individual selections has not been made for the Facility Tier XXX. |
| 17110, 17120 | The maximum number of allowed individual selections has been exceeded for the Facility Tier XXX. |
| 17208        | Citation does not exist (bad UID)  |
| 17202        | CONTRAVENTION_UID missing  |
| 17203        | CONTRAVENTION_UID must be a valid UID  |
| 17204        | IS_SPECIAL_STATUS missing  |
| 17205        | IS_SPECIAL_STATUS must be a valid Boolean  |
| 17300        | The Credit Card Expiration Date XXX is invalid.  |
| 17301        | Profile UID XXX is invalid.  |
| 17302        | Start Date XXX is invalid.   |
| 17303        | End Date XXX is invalid.   |
| 17304        | Credit Card Profiles are not enabled for the Payment Method XXX.                                 |
| 17305        | PERMISSION_PROFILE_RELATIONSHIP_UID XXX is invalid.  |
| 17404        | Invalid facility specified   |
| 17500        | Invalid return reason.   |
| 17502        | Cannot reserve or sell permit XXX. The permit XXXX is not in the user's YYY custody.             |
| 17503        | The permit is on an external payment plan and the distribution type cannot be specified.         |
| 18501        | The permit control group and number range UIDs don't match when creating a bulk permit stack.    |
| 18502        | The expiration date of the bulk permit stack is before the effective date.                       |
| 18503        | The NumberOfPermits is required for non-inventoried permits.                                     |
| 18505        | An insufficient number of inventoried permits remains for the bulk                               |

| Code  | Description   |
|-------|---|
|       | stack.  |
| 18550 | General delete bulk permit error.   |
| 18551 | Delete bulk permit purchasing error.  |
| 23000 | Invalid permission UID.   |
| XXXX  | Required field missing (field name).  |
| XXXX  | Permit is not a value credential.   |
| XXXX  | Permit is expired, cannot add value.  |
| XXXX  | Not enough money to cover value.  |
| XXXX  | Permit is not in active state.  |
| XXXX  | Value Credential does not have value credential fee schedule, please upgrade control group. |
| XXXX  | Not a value credential PCG.   |
| XXXX  | Number Of Units required for value credential.  |
| XXXX  | Third party not related to entity.  |
| XXXX  | Sell on Use to third party not valid for permit (setting on control group).                 |
| XXXX  | Number Of Units must match default value of control group.                                  |
| XXXX  | Non-web services cash drawer is already open.   |

## CHAPTER 6: MISCELLANEOUS NOTES FOR DEVELOPERS

### QUERYING THE T2 FLEX DATABASE

Web services are used to insert and update certain types of records in the T2 Flex database, and to perform complex calculations such as determining the price of a permit under a specific set of sale circumstances. The T2\_Flex\_Misc Web service contains the ExecuteQuery() Web method, which allows you to query data from T2 Flex.

While the ExecuteQuery() Web method is the preferred way for both hosted and self-hosted customers to extract data from a T2 Flex database, self-hosted customers still have the option to connect directly to Oracle and use SQL commands to retrieve information.

**WARNING:** Although retrieving T2 Flex data requires the use of SQL SELECT commands, the use of the SQL commands INSERT, UPDATE, and DELETE commands **is not supported**. Using these commands, instead of Web services, to manipulate the data may cause T2 Flex to function incorrectly, as not all of the constraints on T2 Flex data are enforced by Oracle. Some constraints are enforced only in the T2 Flex Business Object Layer, which is used by Web services. Bypassing the Business Object Layer by directly manipulating T2 Flex data with SQL commands could cause data integrity problems.

Connecting your program to Oracle requires that the Oracle client software be available on any computer where your program will run (or, in the case of a web program, on the web server). Installation of the Oracle client is not discussed in this document. Support for obtaining and installing the Oracle client on Microsoft Windows platforms is available from T2 Technical Support. For information on installing the Oracle client on other platforms, see the documentation provided by Oracle.

The programming steps required to query an Oracle database differ from one programming language to another and are likewise not covered in this document. Consult the documentation or public Internet resources available for your programming environment to learn how to issue SQL commands to an Oracle server in your environment.

In order to query the Oracle database, your program must connect to Oracle using the credentials of a T2 Flex user that has the **External Read Only Data Access** security right (see Section 3.0 of this document).

To aid programmers in querying the T2 Flex database, the following tools have been provided:

#### Data Dictionary

The T2 Flex Data Dictionary is an indispensable reference for developers. The Data Dictionary is a list, in Microsoft Excel spreadsheet format, that contains information

about every column (field) of every table in the T2 Flex database. In addition, the most commonly used database views are described in the Data Dictionary.

To obtain the latest copy of the Data Dictionary, contact T2 Technical Support.

## Entity Relationship Diagram

The T2 Flex Entity Relationship Diagram (ERD) is a graphical representation of the T2 Flex database, illustrating the relationships among database tables.

You can download the latest copy of the ERD from the Documentation Downloads portion of the T2 Systems Website. Contact T2 Support for assistance.

## Querying Custom Fields

Values for all custom fields in the application reside in a single table called CUSTOM\_DATA. This feature of the database design allows for great flexibility in creating as many custom fields as one desires, on many different types of objects (e.g., customers, vehicles, etc).

However, this means that custom fields are not part of the tables they are associated with (for example, a custom field created for customer records will not be found in the ENTITY table). Therefore, it can be difficult for a developer to query the database and determine the value of a custom field for a particular record. To address this problem, two SQL functions have been provided to aid developers in querying for values of custom fields:

### GET\_CUSTOM\_TEXT

The GET\_CUSTOM\_TEXT function returns the value of a custom field for a specific record. The function takes three arguments:

- Abbreviation for table associated with the custom field. Valid abbreviations include:
  - ADJ (Adjudication)
  - CON (Contravention)
  - ENT (Entity)
  - PER (Permission)
  - VEH (Vehicle)
- Name of the custom field. Note that the field name must be used, not the label used to describe that field in T2 Flex.
- UID of the record for which the custom field value is sought.

For example, to find the value of a custom field named “INSURANCE\_CO” for the entity (customer) with UID 3716, one would perform the following query:

```
select GET_CUSTOM_TEXT('ENT','INSURANCE_CO',3716) from dual
```

Or, to find all customers with a value of “25” in a custom field named “YEARS\_OF\_SERVICE”:



```
select ENT_UID from ENTITY where  
GET_CUSTOM_TEXT('ENT','YEARS_OF_SERVICE',ENT_UID) = '25'
```

### GET\_CUSTOM\_LKP

If the custom field is a lookup field, then GET\_CUSTOM\_TEXT will return the UID corresponding to the lookup value selected. If you wish to retrieve the value associated with that UID, then use the function GET\_CUSTOM\_LKP instead.

GET\_CUSTOM\_LKP takes the same arguments as GET\_CUSTOM\_TEXT.

## ERROR LOGGING

Web services can be configured to generate a log file of all errors generated by Web Services. This feature is disabled by default. To enable it, the following configuration tasks must be performed on the web server where the Web services are installed.

- 1) Change the web.config file. A new key “ErrorLog” has been added to the web.config file. This value controls the level of error logging. By default it is set to 0. A list of all possible values appears below

| Error Log Value | Log Level                                       |
|-----------------|---|
| 0               | Off (This is the default setting.)              |
| 1               | Error Message and Error Number                  |
| 2               | Error Message, Error Number, and XmlData String |

- 2) Set the correct permits for the “Logs” folder. This folder is located in the folder where Web Services were installed. The “IIS\_WPG” group should be granted the “Write” permit for that folder.

If this function is not enabled correctly, it will fail silently. The only indication that it is not working will be the absence of a log file.

## CHAPTER 7: KNOWN ISSUES LIST

The following issues with Web Services were known to exist at the time of publication of this document. For each issue described below, if a workaround is available, the workaround is listed along with the issue description.

### T2\_FLEX\_ENTITIES

#### InsertUpdateEmailAddress() v1.0

##### *Issue*

The EMAIL\_ADDRESS parameter is required when updating an e-mail address, even though it should not be. Error 9027 is returned.

##### *Workaround*

If you do not wish to update the value of this parameter, then pass the existing address in the EMAIL\_ADDRESS parameter.

##### *Issue*

The ENTITY\_UID parameter is required when updating an e-mail address, even though it should not be. The .NET Framework returns an error when ENTITY\_UID is omitted.

##### *Workaround*

It is not possible to update the ENTITY\_UID on an existing e-mail address to refer to a different customer. Therefore, pass the UID of the existing customer in the ENTITY\_UID parameter.

### T2\_FLEX\_PERMISSIONS

#### SellPermission() v1.0

##### *Issue*

If T2 Flex returns a price of zero dollars for a permit, and an attempt is made to sell the permit for a different price using the PERMISSION\_SALE\_PRICE parameter, using a payment plan payment method causes an error.

##### *Workaround*

Configure T2 Flex such that a price greater than zero dollars is returned as the default price. Then override that default price with the actual price.

##### *Issue*

Error 2271 may be returned when selling a PermitDirect permit.

**Workaround**

This error indicates that the permit was sold successfully, but the order fulfillment request could not be sent automatically to the PermitDirect server. Consider ignoring this error code in your programming. The request can be sent manually to the PermitDirect server by a T2 Flex user.

**T2\_FLEX\_VEHICLES****InsertUpdateVehicle() v1.0****Issue**

If a new vehicle make has been added to T2 Flex, Web services will not be able to determine the model, and attempts to insert or update a vehicle using this make will fail if a model value is not passed. The error code returned is 9106, with message text “General web method error- PowerPark Flex Error: VehicleModelLookup with ID of xxxx does not exist in the database.”

**Workaround**

Look up the UID for the default model belonging to the make, and pass that UID in the VEHICLE\_MODEL parameter.