

APPENDIX 13 – EXTERNAL SYSTEM INTERFACES

Appendix 13.1: Roadside Interface Control Document & Roadside Interface Guidelines

Appendix 13.2: SR 167 TCS – to – SIMMS ICD

Appendix 13.3: Sample of TMC Data in XML Format

See: <http://images.wsdot.wa.gov/nwflow/wireless/SR167TrafficDataNWNew.xml>

This page intentionally left blank.

APPENDIX 13.1 – ROADSIDE INTERFACE CONTROL DOCUMENT & ROADSIDE INTERFACE GUIDELINES

NOTE: THE ROADSIDE INTERFACE GUIDELINES MAY CHANGE BETWEEN THE RFP RELEASE AND DESIGN. A FINAL VERSION WILL BE PROVIDED TO THE VENDORS FOR DESIGN PURPOSES.

This page intentionally left blank.

WSDOT

Roadside

Interface Control Document

Version 1.12.3
April 7, 2011



The *rite*
Solution

Created for:

Washington State Department of Transportation

Contract Number K397

ETC / Electronic
Transaction
Consultants

1705 North Plano Road Richardson, Texas USA 75081 tel: 214-615-2302, fax: 214-615-5001 www.etcc.com

Document Control

Date	Revision Version	Author	Summary of Changes
12/30/2009	0.1	Niyi Olajide	Initial Draft
06/19/2010	0.2	Will Kooiman	Second Draft
06/23/2010	0.3	Dave Ricketts	Per ETCC request, made minor changes to support distribution to all Teams (WSDOT, Telvent Caseta, and TransCore). Also added "ETCC" prefix to document name.
08/10/2010	1.0	Jun Fu, Dan Tian, Will Kooiman	<ol style="list-style-type: none"> 1. Changed incremental TVLs to file transfers 2. Removed ROI Coordinates. Our preferred method of handling ROI images is to provide a separate image file for the ROI. 3. Added entry date and entry lane information for HOT lane support. 4. Removed confidential and proprietary verbiage from the footer. WSDOT might want to consider adding their own confidentiality verbiage. 5. Added tables to each service to clarify which fields are mandatory and which are optional. 6. Made various updates per reviewer comments. 7. Added Appendix for XSD and WSDL. 8. Modify Appendix E with detailed error messages 9. Add Frequency in Section 2.3 Data Exchange table 10. Add Entry Facility/Plaza/Lane and TransactionDate fields to support SR 167 HOT transactions. 11. Removed login service.
08/20/2010	1.1	Will Kooiman	<p>Removed Userid/Password. Authentication will be with x.509 digital certificate.</p> <p>Removed request to Telvent Caseta to reduce image resolution.</p>
09/04/2010	1.2	Will Kooiman	Addressed all but 11 comments from the review of the document. The remaining 11 comments will be addressed in the next version.
09/09/2010	1.3	Dan Tian	Updated the Rate Schedule section based on ETCC Illinois design.
09/22/2010	1.4	Dan Tian /	Addressed all but 4 comments from the previous review of the document. See the updated comments list for

Date	Revision Version	Author	Summary of Changes
		Will Kooiman	details.
10/18/2010	1.5	Dan Tian	<ol style="list-style-type: none"> 1) Added the Ack / Nak interface for TVL / Rate Schedule; 2) Changed the time format from "99:99" to "23:59"; 3) Modified the LaneModes definitions
11/4/2010	1.6	T. Owen	<ol style="list-style-type: none"> 1) Updated Appendix F, AgencyCodes, FacilityCodes, PlazaCodes, TransactionTypes, and samples for transactions and Image files. Updated business continuity with solution for long term network outages.
11/10/2010	1.7	Dan Tian	<ol style="list-style-type: none"> 1) Updated Appendix C – AcknowledgeFile XML 2) Added Appendix C - AcknowledgeFile WSDL 3) Updated Appendix F.11 - Transponder Status 4) Updated the Acknowledgement Web Service Section, including adding "FacilityCodes"
11/11/10	1.8	C.Webster	<ol style="list-style-type: none"> 1) Addressed final comments from 09/21/2010 2) Addressed final comments from Ver 1.4
11/18/2010	1.9	Dan Tian/T.Owen	<ol style="list-style-type: none"> 1) TVL CSC Class to Optional 2) Updated Appendix F.1 AckCodes 3) F.4 Facility Codes 4) F.5 Interface Names 5) F.6 Lane Modes 6) F.7 Plaza Codes 7) F.9 Transaction Types 8) F.10 Transponder Protocols 9) F.12 UO Codes
11/23/2010	1.10	T.Owen	<ol style="list-style-type: none"> 1) Change Toll Transaction data element – ReverseDirectionFlag to Optional
11/23/2010	1.10.1	Dan Tian	<ol style="list-style-type: none"> 1) Change CSCClassification to Optional in Section 4.3.3
12/4/2010	1.11	T.Owen	Updated per WSDOT Comments with Change Tracking.
12/6/2010	1.12	T. Owen	Updated per Comments/Coordination Meeting with Tracking. Updated Image Example, Corrected OCRStateConfidence, deleted OCRConfidence.

Date	Revision Version	Author	Summary of Changes
12/7/2010	1.12.1	Dan Tian	Changed VehicleClass to AVClassification
12/12/2010	1.12.2	T. Owen	Revised 4.3.1.3 wording for non-compliant transactions
04/07/2011	1.12.3	C. Webster	SelectedImageName and SelectedROImageName made optional Maximum transponder id increased to 2147483647

Contents

1	Introduction	1
1.1	Overview.....	1
1.2	Purpose	1
1.3	Definitions.....	1
1.4	References	2
1.5	Placeholder Values.....	2
2	General Information.....	3
2.1	Conflicting / Changed Requirements	3
2.2	Data Exchange Format	4
2.3	Transfer Protocol.....	4
2.4	Data Exchanges	4
3	XML Formatting.....	6
3.1	Built-in Data Types	6
3.2	Complex Types	6
4	Web Services.....	8
4.1	Processing Guidelines.....	8
4.2	Security.....	9
4.2.1	Authentication.....	9
4.2.2	Sensitive Information	9
4.3	Toll Transaction	9
4.3.1	Description.....	9
4.3.2	Data Flow.....	10
4.3.3	Data Elements	11
4.3.4	Request Example	12
4.4	Daily Reconciliation	14
4.4.1	Description.....	14
4.4.2	Data Elements	15
4.4.3	Request Example	15
4.5	Acknowledgement Web Service	16
4.5.1	Acknowledgement Format.....	16
4.5.2	Acknowledgement Ack Code and Ack Message.....	17
4.5.3	Acknowledgement XSD.....	19
5	File Transfers	21
5.1	Overview.....	21
5.2	TCS File Server.....	21

Contents

5.2.1	TCS File Server Security	21
5.3	Transponder Validation List (TVL)	22
5.3.1	Overview	22
5.3.2	Transfer Protocol	22
5.3.3	File Archival	22
5.3.4	Compression.....	22
5.3.5	File Types	22
5.3.6	Naming Convention	23
5.3.7	Acknowledgement	23
5.3.8	Directory Structure.....	23
5.3.9	Data Elements	23
5.3.10	File Example.....	24
5.4	Rate Schedule	24
5.4.1	Overview	24
5.4.2	Frequency	24
5.4.3	Transfer Protocol	25
5.4.4	File Archival	25
5.4.5	File Type	25
5.4.6	Naming Convention	25
5.4.7	Acknowledgement	25
5.4.8	Directory Structure.....	25
5.4.9	File Format.....	25
5.4.10	Record Format	27
5.5	Images	29
5.5.1	Overview	29
5.5.2	Transfer Protocol	29
5.5.3	File Archival	29
5.5.4	Compression.....	29
5.5.5	Naming Convention	29
5.5.6	Directory Structure.....	30
5.5.7	Regions of Interest (ROI).....	30
6	Operations	32
6.1	Scheduled Maintenance.....	32
6.2	Unscheduled Outages.....	32
6.3	Change Management.....	33
6.4	Troubleshooting.....	33
6.5	Disaster Recovery / Business Continuity	33
6.6	Contact Information	33

Contents

Appendix A Toll Transaction Web Service	34
A.1 XSD	34
A.2 WSDL	39
Appendix B Daily Reconciliation Web Service	40
B.1 XSD	40
B.2 WSDL	42
Appendix C AcknowledgeFile	43
C.1 XSD	43
C.2 WSDL	43
Appendix D TransponderValidationList	45
D.1 XSD	45
Appendix E Rate Schedule Sample File	47
Appendix F Domains	51
F.1 AckCodes	51
F.2 TransponderAgencyCodes.....	53
F.3 DeclarationStates	53
F.4 FacilityCodes	53
F.5 InterfaceNames	53
F.6 LaneModes.....	54
F.7 PlazaCodes	54
F.8 TCSStatusCodes.....	55
F.9 TransactionTypes	55
F.10 TransponderProtocols	55
F.11 TransponderStatusCodes	55
F.12 UnusualOccurrenceCodes	56
F.13 VehicleClassTypes.....	56
Appendix G Complex Types	57
G.1 TransponderType	57
G.2 ImageListType.....	57
Appendix H Response Codes	59

1 Introduction

1.1 Overview

This document is a communication tool designed to manage expectations, improve communications, and clarify information required to use the interfaces between the Washington State Department of Transportation (WSDOT) Toll Collection Systems (TCS) and the Washington State Department of Transportation (WSDOT) Customer Service Center (CSC) Back Office. It includes technical details and operational information necessary to ensure proper usage and support for service operation.

This document contains the interface contract for the WSDOT TCS to CSC interfaces. Changes to the specification and or agreements contained in this document must be approved and subsequently communicated via update to this document. Updated versions of this document must be distributed to all consumers and support staff for the services described herein.

1.2 Purpose

The purpose of this document is to describe the data exchanges between the WSDOT TCS and the CSC Back Office.

1.3 Definitions

Word / Acronym	Definition
CSC	Customer Service Center
ICD	Interface Control Document
TCS	Toll Collection System
XSD	XML Schema Definition
WSDL	Web Service Description Language
Tag	A Tag is a markup construct that is used in XML documents. It begins with "<" and ends with ">".
Markup and Content	XML documents consist of markup and content. Markup begins with the character "<" and ends with a ">". Strings of characters which are not markup are content.
Element	A logical component of an XML document which begins with a start-tag and ends with a matching end-tag. The characters between the start- and end-tags, if any, are the element's <i>content</i> , and may contain markup, including other elements, which are called <i>child elements</i> . An example of an element is: <AccountID>12345</AccountID>
XML Document	This ICD defines data structures in XML format for the purpose of passing data from one system to another. The data structures are called "XML Documents"
Facility	A toll facility is usually a separately funded construction project. Examples of toll facilities are the Tacoma Narrows Bridge, the SR520 bridge, and the SR167 HOT lanes.
Plaza	A plaza traditionally includes a physical structure with gates, lanes, toll collectors, coin baskets, unattended lanes, and so on.

	Open Road Tolling (ORT) roadways have no need for physical plaza structures. Instead, they have tolling zones with minimal structures to mount electronic toll collecting equipment. The term <i>plaza</i> is sometimes used interchangeably with <i>tolling zone</i> , even though there is no physical plaza with ORT.
Lane	A lane is a single lane of traffic at a toll collecting point.
TCS	A Toll Collection System refers to all of the electronic toll collecting equipment at a facility.
TCS File Server	The file server provided by the TCS vendor to support file transfers between the TCS and the CSC Back Office
CSC Back Office	The Customer Service Center back office. This consists of all interfaces in the Customer Service Center that is not part of the user interface.

1.4 References

The following documents were referenced in writing this ICD:

- WSDOT Statewide CSC – Appendix 2 Scope of Work (SOW), V3.0
- SR520 TCS RFP – Appendices 3 & 4, dated 05/14/2010
- Tacoma Narrows / SR 167 Hot Lane Integration Interface Control Document, V2.0
- Washington State ISB Policies and Standards (located at <http://isb.wa.gov/policies/Default.aspx>):
 - Policy No. 401-S4, IT Security Standards
 - Policy No. 704-S1, IT Standards & Protocol Directions
 - Policy No. 801-S2, State Standards for Internet Markup Language (HTML)

1.5 Placeholder Values

Some Example values in this document are placeholders only. Actual Values for testing and implementation will be provided to stakeholders Ad hoc as development test and implementation require.

2 General Information

2.1 Conflicting / Changed Requirements

Section 34.2.3 of the CSC SOW discusses the requirements for the interfaces between the CSC and the TCS systems. It requests that the interfaces be the same to the extent possible. We believe this ICD has accomplished that goal.

Throughout the design, we discovered inconsistencies with the requirements. These are summarized below:

Topic	SR167 / TNB	SR520	This ICD
Transaction Upload	Transactions are uploaded in batch via FTP	Transactions are uploaded in batch via FTP	Transactions are uploaded near real-time via web services
Image File Naming	No images – does not apply.	Complex image file names including 7 data elements.	Industry acceptable file names including key data elements that uniquely identify a transaction.
XML Tag Names	Tag names are in all uppercase separated by underscores.	No mention of XML.	Tag names are in upper/lower case without underscores. Data element names are industry standard as much as possible.
Check Files	“Check” files are generated to denote that transfers are complete. This is an industry standard practice when transferring files via FTP.	No mention of check files.	Instead of using check files, files will be transferred to a <i>working</i> directory. When the transfer is complete, it will be moved to a <i>ready</i> directory. This is also an industry standard practice.
Acknowledgement Files	Acknowledgment files are generated to acknowledge the receipt of transferred files.	No mention of acknowledgement files.	Acknowledgements are done via web services.
File Transfer	Files are transferred via	FTP?	Files are transferred

Method	FTP.		via SFTP.
TVL Transfer Frequencies	TVL files are transferred twice daily.		Full TVL files are generated once daily. Incremental TVL files are generated hourly (23 per day).
Daily Reconciliation	No daily reconciliation	Daily reconciliation file must be sent once daily, but details are not specified.	Daily reconciliation is done via a web service.

2.2 Data Exchange Format

All data, with the exception of images, will be specified in XML format as defined in the Second Edition of XML Schema documented at: <http://www.w3.org/TR/xmlschema-0/>

Images will be transmitted in Jpeg format, since it is used exclusively by the tolling industry. TCS vendors must use a version of Jpeg that displays properly in the following browsers:

- Microsoft Internet Explorer 7 or higher
- Mozilla Firefox Version 3.6 or higher
- Google Chrome 5.0 or higher

2.3 Transfer Protocol

The preferred message protocol is web services.

Bulk data and images will be transmitted using SFTP file transfer architecture.

2.4 Data Exchanges

The following data exchanges are specified in this document:

Data Exchange	Format	Push / Pull	Method / Protocol	Frequency
Toll Transaction	XML	TCS Push to CSC	Web Service / HTTPS	Continuously – 24/7
Transponder Validation List	XML	CSC Push to TCS	File Transfer / SFTP	Full – Once per day, prior to 4am PST Incremental – Hourly, excluding the period from 3am to 4am PST, 23 total per day
Daily Reconciliation Report	XML	TCS Push to CSC	Web Service / HTTPS	Once per day

Images	Jpeg	CSC Pull from TCS	File Transfer / SFTP	Continuously – 24/7
Rate Schedule	TXT	CSC Push to TCS	File Transfer / SFTP	As-Needed
File Acknowledgement	XML	TCS Push to CSC	Web Service / HTTPS	As-Needed with File Download

3 XML Formatting

XML documents in this interface follow XML formatting rules as defined in the Second Edition of XML Schema documented at: <http://www.w3.org/TR/xmlschema-0/>

3.1 Built-in Data Types

XML includes data types called built-in data types. The following built-in data types are used in this ICD:

- int
- long
- decimal
- boolean
- date
- time
- dateTime.

The built-in data type for date, dateTime, and time allows timezone to be specified. For this interface all date elements will be specified without a timezone and will be in the local timezone for Seattle, Washington.

The dateTime built-in type is:

Specific instant of time. ISO 8601 extended format

CCYY-MM-DDThh:mm:ss.

Example, to indicate 1:20 pm

on May the 31st, 1999 for Eastern Standard Time which

is 5 hours behind Coordinated Universal Time (UTC):

1999-05-31T13:20:00-05:00.

3.2 Complex Types

XML allows the definition of complex data types. A basic complex data type extends a simple type by specifying valid values or ranges.

Here is an example of a complex type that limits the values to "1" and "2". The base type is xsd:string. The values are limited with the enumeration keywords.

```
<xsd:simpleType name="TransactionTypeType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="1"/>
    <xsd:enumeration value="2"/>
  </xsd:restriction>
</xsd:simpleType>
```

Complex types may also be used to specify groups of data elements (traditionally called records).

Here is an example of complex type that defines a record:

```
<xsd:complexType name="USAddress" >  
  <xsd:element name="name" type="xsd:string"/>  
  <xsd:element name="street" type="xsd:string"/>  
  <xsd:element name="city" type="xsd:string"/>  
  <xsd:element name="state" type="xsd:string"/>  
  <xsd:element name="zip" type="xsd:decimal"/>  
</xsd:complexType>
```

Complex types are discussed in detail at <http://www.w3.org/TR/xmlschema-0/>.

4 Web Services

4.1 Processing Guidelines

Web Services will be provided by the CSC Back Office. The TCS calls the CSC Web Services to push data to the CSC Back Office or send out acknowledgement to CSC. Web Services are accessed through an HTTPS connection over an IPsec VPN.

An IPsec VPN is used to link two networks over an insecure channel (the Internet). It provides a secure "tunnel" from one network to the other.

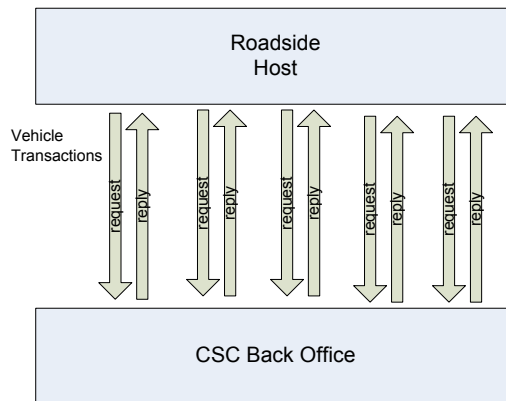
HTTPS is used to secure communications over any network, whether the network is internal (intranet) or external (Internet).

While it is not required (with Category 2 data) to encrypt during transmission, we prefer using HTTPS whenever possible. It is more secure, and it is easier to manage from a security standpoint.

Using HTTPS should not introduce additional complexity. It will, however, introduce extra processing. The processing load will be determined during the testing phase, at which point the architecture might be modified to address issues. If testing proves the extra processing is an issue, we will look at alternatives.

Each Web Service transaction consists of a request and a reply in a synchronous session. If the request is successful, the reply message will include a status code that denotes success. If the request is not successful, the reply message will include an error message.

The following drawing shows typical web service transitions:



Only one transaction is sent per request. There is no batching of requests. So, for example, if the TCS needs to upload 1,000 toll transactions, there will be 1,000 toll transactions for a total of 1,000 requests and 1,000 replies.

Transaction http request capacity at the roadside Hosts has been calculated based upon current daily traffic counts as a baseline with a contractually required potential of 500,000 transactions per day. Each TCS vendor will be initially limited to 10 parallel threads which at the http request side has the

capacity to provide conservatively 10 requests per second with each request being considered in industry terms as “trivial workload request” (this does not mean these requests are all trivial per se they begin as a simple request and can branch into complex modes relatively quickly based on the user interaction.) The TCS vendor’s http request systems can generate over a hundred http requests per second when parsing from a circular file or a FIFO file mechanism. Since the format and content of each of these http requests is defined by this ICD it is also possible to fairly accurately predict the loading of the web server at the CSC Side. The web services server is virtualized and can be a single web service or a virtual farm of web servers with load balancing between each virtual server providing the ability to scale out and up with no rewrite of any part of the web services interface.

4.2 Security

4.2.1 Authentication

Authentication will be implemented via Digital Certificate. ETC will generate self-signed x.509 certificates and will distribute the public key to the TCS vendors.

4.2.2 Sensitive Information

Sensitive information is protected as specified in Washington State ISB Security Standard – Policy No. 401-54. This standard specifies that data must be classified, and properly protected.

All data transmissions between the TCS Vendor and the CSC Back Office is classified as Category-2. The justification for this classification is below:

- SR520-4.205 - WSDOT has classified image data as category 2.
- SR520-4.48 - Images are limited to the vehicle and vehicle license plate only (for more information see <http://apps.leg.wa.gov/rcw/default.aspx?cite=46.63.160>)
- In a previous meeting with the DIS, ETC was told that Toll Transaction data flows and image data flows are Security Category 2.
- No credit card information is passed between the TCS Vendor and the CSC Back Office
- No personally identifiable information is passed between the TCS Vendor and the CSC Back Office

If the security category changes, this ICD will need to change appropriately.

4.3 Toll Transaction

4.3.1 Description

The purpose of the TollTransaction web service is to submit toll transactions to the CSC Back Office for posting. The transaction request will contain all required transaction data occurring at the lane so that the CSC Back Office can apply the transactions to the appropriate customer account.

Toll transactions will be sent from all lane types (HOT, ORT, mixed use) that require posting to a customer account. Cash transactions, which are not posted to an account, will not be sent.

Transactions should be sent in a timely manner. If the transactions fall behind by more than 4 hours, the TCS vendor should contact the CSC help desk to remediate. Per CSC business rules, the CSC will reject toll transactions that are older than 60 days.

Transactions that cannot be posted to accounts will result in the facility not receiving the amount due for the transaction.

4.3.1.1 Transponder Enforced

Transponder enforced transactions must include transponder information and may optionally include images. Some facilities are required to provide images for all transactions. Others are required to provide images for low-balance transponders only.

4.3.1.2 Photo Enforced

Photo Enforced transactions must include at least one image. They may also include transponder information, for example when a transponder was read, but the transponder status was invalid.

When image(s) are included, the Selected Image should also be included along with all OCR and ROI information.

4.3.1.3 Missing Transponder and Photo

If there are no valid transponder or images, the TollTransaction cannot be posted to a CSC account. However, All transactions without transponder or image will be sent by Roadside facilities up to CSC in order to capture the transactions for counts and reconciliation among systems. The transaction count will also be included in the DailyReconciliation web service.

4.3.2 Data Flow

Upon receipt of the web service transaction, the CSC Back Office will perform the following data validation checks:

- Required data elements have been specified
- Data elements are of the proper data type and range as specified in the XSD
- TransactionDateTime > current date - 61 days
- No duplicate transaction. The combination of FacilityCode, PlazaCode, LaneNumber, TransactionDateTime, and SequenceNumber guarantees uniqueness for transactions at the lane.

If the transaction passes data validation checks, it will be accepted, and a success code will be returned. It will be stored in the CSC database and queued for later posting to the customer's account.

If the transaction does not pass data validation checks, it will be rejected, and a response code will be returned. It will be stored in the CSC database, but it will not be posted to the customer's account. It will be retained in a failed transactions table for later research and/or reporting.

Whether the transaction passes or fails data validation checks, it will be committed to the CSC database before the web service sends an acknowledgement. The transactions that pass data validation will be posted to the appropriate account. Those that fail will be retained in a "failed transactions" table, but they will not be posted to a customer's account.

4.3.3 Data Elements

The following data elements are accepted in the TollTransaction:

TagName	Required?	Data Type	Notes
FacilityCode	Required	String	Identifies the tolling facility. For HOT transactions, this is the Exit Facility <i>Domain: FacilityCodes</i>
PlazaCode	Required	String	Code assigned to the plaza or tolling zone. For HOT transactions, this is the Exit Plaza <i>Domain: PlazaCodes</i>
LaneNumber	Required	Int	Identifies the lane within the plaza (or tolling zone). The lane number must be a valid lane number for the specified plaza (or tolling zone). It will be validated before the transaction is posted to the appropriate CSC account. For HOT transactions, this is the Exit Lane Number. <i>Range: 1 to 99</i>
TransactionDateTime	Required	dateTime	Toll Transaction dateTime in the local time zone. Fractions of seconds are required. TCSs that do not support fractional seconds should pass zeros for the fractional part.. Cannot be in the future or more than 60 days in the past. For HOT transactions, this is the date/time the vehicle exited the facility.
EntryFacilityCode	Optional	String	Identifies the entry facility code for HOT transactions. Required for HOT transactions. It is not used for non-HOT transactions <i>Domain: FacilityCodes</i>
EntryPlazaCode	Optional	String	Identifies the entry plaza for HOT transactions. Required for HOT transactions. It is not used for non-HOT transactions <i>Domain: PlazaCodes</i>
EntryLaneNumber	Optional	Int	Identifies the entry lane number for HOT transactions. Required for HOT transactions. It is not used for non-HOT transactions <i>Range: 1 to 99</i>
EntryTransactonDateTime	Optional	dateTime	Identifies the date/time the vehicle entered the facility for HOT transactions. Required for HOT transactions. It is not used for non-HOT transactions. Fractions of seconds are required.
SequenceNumber	Required	Long	This number along with PlazaCode, FacilityCode, LaneNumber, and

			TransactionDateTime guarantees uniqueness for transactions at the lane. The SequenceNumber can be generated by any algorithm as long as it guarantees uniqueness according to the previously mentioned rules. It is acceptable for the sequence number to skip numbers.
TransactionType	Required	String	Identifies the transaction type, photo-enforced or transponder-enforced. <i>Domain: TransactionTypes</i>
AVCClassification	Required	String	Vehicle classification as identified by Automatic Vehicle Classification system in the lane. <i>Domain: VehicleClassTypes</i>
CSCClassification	Optional	String	Vehicle classification as identified in the Transponder Validation List. <i>Domain: VehicleClassTypes</i>
Transponder	Optional	TransponderType	This is a complex type that includes all fields that describe a transponder. The complex type is defined in Appendix G.1. This data element is required for transponder-enforced transactions, otherwise it is optional. If the TCS reads a transponder, even if the transponder is invalid, it should be submitted in the transaction.
ImageList	Optional	ImageListType (see details in Appendix G.2)	Required under these conditions: a) Photo-Enforced Transaction b) The facility is required to provide images for all transactions c) The facility is required to provide images for low-balance transactions.
FareAmount	Optional	decimal	Required for HOT Lanes. <i>Range: 0.00 to 999.99</i>
VehicleSpeed	Optional	int	Vehicle speed identified by the lane equipment in MPH. <i>Range: 0 to 999</i>
UnusualOccurrenceCode	Optional	string	Unusual occurrences attributed to the system. <i>Domain: UnusualOccurrenceCodes</i>
TCSStatusCode	Required	string	Describes the status of the toll collecting equipment. <i>Domain: TCSStatusCodes</i>
ReverseDirectionFlag	Optional	boolean	Reverse direction indicator. A true value indicates the traffic was traveling in the reverse direction.
LaneMode	Required	string	<i>Domain: LaneModes</i>
AxleCount	Optional	int	<i>Range: 0 to 15</i>

4.3.4 Request Example

```
<?xml version="1.0"?>
<soap:envelop>
```

```

<soap:body>
<sendTransactionData>
  <TollTransaction>
    <FacilityCode>          SR16    </FacilityCode>
    <PlazaCode>            TNB      </PlazaCode>
    <LaneNumber>           1        </LaneNumber>
    <TransactionDateTime> 2010-07-27 11:32:29.123456
  </TransactionDateTime>
  <SequenceNumber>       12345    </SequenceNumber>
  <TransactionType>      1         </TransactionType>
  <AVCClassification>    2         </AVCClassification>
  <CSCClassification>   2         </CSCClassification>

  <Transponder>
    <TransponderID>       1234567890 </TransponderID>
    <IssuingAgencyCode>  1         </IssuingAgencyCode>
    <TransponderProtocol> SeGO      </TransponderProtocol>
    <Status>              1         </Status>
    <DeclarationState>    1         </DeclarationState>
    <LowBattery>          false     </LowBattery>
  </Transponder>

  <FareAmount>           2.50     </FareAmount>
  <VehicleSpeed>        67        </VehicleSpeed>
  <TCSStatus>           1         </TCSStatus>
  <ReverseDirectionFlag> false    </ReverseDirectionFlag>
  <LaneMode>            1         </LaneMode>
  <AxleCount>           2         </AxleCount>

  <ImageList>
    <ImageLocation>      20100727/01/01 </ImageLocation>
    <SelectedImageName>  BR549_A01.JPG </SelectedImageName>
    <SelectedROIImageName> BR549_X01.JPG </SelectedROIImageName>
    <Image>
      <ImageName>         BR549_A01.JPG </ImageName>
      <ImageType>         BSI          </ImageType>
      <OCRPlateNumber>    ACN310      </OCRPlateNumber>
      <OCRPlateState>     WA          </OCRPlateState>
      <OCRPlateType>      1           </OCRPlateType>
      <OCRConfidence>     99          </OCRConfidence>
      <OCRPlateNumberConfidence> 99
      </OCRPlateNumberConfidence>
      <OCRPlateStateConfidence> 99
      </OCRPlateStateConfidence>
      <OCRPlateTypeConfidence> 99
      </OCRPlateTypeConfidence>
    </Image>
    <Image>
      <ImageName>         BR549_B01.JPG </ImageName>
      <ImageType>         BSI          </ImageType>
      <OCRPlateNumber>    ACN310      </OCRPlateNumber>
      <OCRPlateState>     WA          </OCRPlateState>
      <OCRPlateType>      1           </OCRPlateType>
      <OCRConfidence>     99          </OCRConfidence>

```

```

        <OCRPlateNumberConfidence> 99
        </OCRPlateNumberConfidence>
        <OCRPlateStateConfidence> 99
        </OCRPlateStateConfidence>
        <OCRPlateTypeConfidence> 99
        </OCRPlateTypeConfidence>
    </Image>
    <Image>
        <ImageName> BR549_X01.JPG </ImageName>
        <ImageType> ROI </ImageType>
        <OCRPlateNumber> ACN310 </OCRPlateNumber>
        <OCRPlateState> WA </OCRPlateState>
        <OCRPlateType> 1 </OCRPlateType>
        <del><OCRConfidence> 99 </OCRConfidence></del>
        <OCRPlateNumberConfidence> 99
        </OCRPlateNumberConfidence>
        <OCRPlateStateConfidence> 99
        </OCRPlateStateConfidence>
        <OCRPlateTypeConfidence> 99
        </OCRPlateTypeConfidence>
    </Image>
    <Image>
        <ImageName> BR549_X02.JPG </ImageName>
        <ImageType> ROI </ImageType>
        <OCRPlateNumber> ACN310 </OCRPlateNumber>
        <OCRPlateState> WA </OCRPlateState>
        <OCRPlateType> 1 </OCRPlateType>
        <del><OCRConfidence> 99 </OCRConfidence></del>
        <OCRPlateNumberConfidence> 99
        </OCRPlateNumberConfidence>
        <OCRPlateStateConfidence> 99
        </OCRPlateStateConfidence>
        <OCRPlateTypeConfidence> 99
        </OCRPlateTypeConfidence>
    </Image>
</ImageList>
</TollTransaction>
</sendTransactionData>
</soap:body>
</soap:envelope>

```

4.4 Daily Reconciliation

4.4.1 Description

The Daily Reconciliation web service will be executed by each TCS once a day to provide reconciliation information to the CSC back office.

The summary level information will be compared to the data received by the CSC back office in a CSC report. The discrepancies will be highlighted to help identify missing data. For example, if the daily reconciliation web service states that the TCS vendor transmitted 5,000 transactions, but the CSC can only find 1,000 transactions, there are apparently 4,000 transactions missing. In this case, the CSC

vendor will work with the TCS vendor to not only find the missing transactions, but also find the cause for the exclusion.

The Daily Reconciliation web service includes the FromDateTime and ToDateTime fields to indicate that the counts provided by the TCS are for the transmission period indicated by these two data fields. Therefore, the CSC will use this date and time range and counts to reconcile with the applicable TCS facility..

The TCS vendors should send summary information for all lanes, transaction types, and AVC classifications, even if the transaction counts are zero. This is needed to ensure that all data has been reported.

4.4.2 Data Elements

TagName	Required?	Data Type	Notes
FromDateTime	Required	dateTime	starting date/time of the reporting period
ToDateTime	Required	dateTime	ending date/time of the reporting period
ArrayList	Required	ComplexType	
ArrayElement minOccurs="1" maxOccurs="unbounded"			
FacilityCode	Required	string	Identifies the tolling facility. <i>Domain:</i> FacilityCodes
PlazaCode	Required	string	Code assigned to the plaza or tolling region. <i>Domain:</i> PlazaCodes
LaneNumber	Required	int	Identifies the lane within the plaza (or tolling zone). This number typically starts with 1, and increments by 1 for each lane in the plaza. <i>Range:</i> 1 to 99
TransactionType	Required	string	<i>Domain:</i> TransactionTypes
AVClassification	Required	string	Vehicle classification as identified by Automatic Vehicle Classification system in the lane. <i>Domain:</i> VehicleClassTypes
TransactionCount	Required	long	Count of transactions

4.4.3 Request Example

```
<soap:envelop>
<soap:body>
<sendDailyReconciliation>
  <DailyReconciliation>
    <FromDateTime> 2010-07-27 00:00:00 </FromDateTime>
    <ToDateTime> 2010-07-28 00:00:00 </ToDateTime>

    <ArrayList>
      <ArrayElement>
        <FacilityCode> TNB </FacilityCode>
        <PlazaCode> 1 </PlazaCode>
        <LaneNumber> 1 </LaneNumber>
        <TransactionType> 1 </TransactionType>
      </ArrayElement>
    </ArrayList>
  </DailyReconciliation>
</sendDailyReconciliation>
</soap:body>
</soap:envelop>
```

```

    < AVCClassification > 1 </ AVCClassification >
    <TransactionCount> 500 </TransactionCount>
  </ArrayElement>
  <ArrayElement>
    <FacilityCode> TNB </FacilityCode>
    <PlazaCode> 1 </PlazaCode>
    <LaneNumber> 1 </LaneNumber>
    <TransactionType> 1 </TransactionType>
    < AVCClassification > 2 </ AVCClassification >
    <TransactionCount> 700 </TransactionCount>
  </ArrayElement>
  <ArrayElement>
    <FacilityCode> TNB </FacilityCode>
    <PlazaCode> 1 </PlazaCode>
    <LaneNumber> 1 </LaneNumber>
    <TransactionType> 1 </TransactionType>
    < AVCClassification > 3 </ AVCClassification >
    <TransactionCount> 900 </TransactionCount>
  </ArrayElement>
  <ArrayElement>
    <FacilityCode> TNB </FacilityCode>
    <PlazaCode> 1 </PlazaCode>
    <LaneNumber> 2 </LaneNumber>
    <TransactionType> 1 </TransactionType>
    < AVCClassification > 1 </ AVCClassification >
    <TransactionCount> 300 </TransactionCount>
  </ArrayElement>
</ArrayListElement>
</DailyReconciliation>
</sendDailyReconciliation>
</soap:body>
</soap:envelope>

```

4.5 Acknowledgement Web Service

The CSC will provide a web service to accept acknowledgement from TCS. The intent of this web service is to provide a generic Ack / Nak service for all roadside interfaces, including TVL and Rate Schedule.

4.5.1 Acknowledgement Format

Field/Element	Data Type	Required	Notes
AckData/			
AckCode	xs:int	Y	Code of Ack returned. (example: 0, 1, 2, etc...) Valid Range: 0 .. 999
AckMessage	xs:string	Y	The actual Ack message. (example: "File Name is missing", "No file is received in the past hour", etc...) Valid Range: String
FacilityCodes	xs:string	Y	Indicated the TCS facility that sends the request. Valid Range: See Appendix F.4

Field/Element	Data Type	Required	Notes
Date	Xs:string	N	It indicates the Date when no TVL full file is received Valid Range: YYYYMMDD (Example: 20101025)
FileName	xs:string	N	This is the same file name in the file sent out to TCS. This field is used to match an acknowledgement to the file sent out. Valid Range: -For TVL Full File: YYYYMMDDHHMMSS_Full_TVL_XML.ZIP -For TVL Incremental File: YYYYMMDDHHMMSS_Full_TVL_XML.ZIP -For Rate Schedule File: YYYYMMDDHHMMSS_Rate_Schedule.txt
InterfaceName	xs:string	Y	It indicates which interface, either TVL or Rate Schedule. Valid Range: "TVL", or "RATE"
TimePeriod	Xs:string	N	It indicates the TimePeriod when no TVL incremental file is received. Valid Range: HH:MM – HH:MM (Example: 06:00 – 07:00; or 06:00 – 09:00)

4.5.2 Acknowledgement Ack Code and Ack Message

Ack Code	Ack Message
0	File Received Successfully
1	No Rate Schedule file is received.
2	No TVL Full File is received
3	No TVL Incremental File is received
11	File cannot be unzipped
12	File cannot be processed
13	File format is not correct
101	The record is blank

102	The TransponderValidationList is missing
103	The FileGenerateTime is missing
104	The TotalRecordCount is missing
105	The CSCClassification is missing
106	The TransponderID is missing
107	The TransponderAgencyCode is missing
108	The TransponderStatusCode is missing
109	The total record count does not match the Header
201	The record is blank
211	The [FareStructure] section is wrong
212	The [FareStructure] – FareID is wrong
213	The [FareStructure] – Fare is wrong
214	The [FareStructure] – class is wrong
215	The [FareStructure] – payment-type is wrong
216	The [FareStructure] – fare-amount is wrong
217	The [FareStructure] – per-axle-fare is wrong
221	The [Schedule-Fare] section is wrong
222	The [Schedule-Fare] – ScheduleID is wrong
223	The [Schedule-Fare] – Period is wrong
224	The [Schedule-Fare] – start-day is wrong
225	The [Schedule-Fare] – start-time is wrong
226	The [Schedule-Fare] – end-day is wrong

227	The [Schedule-Fare] – end-time is wrong
228	The [Schedule-Fare] – fare-id is wrong
231	The [Plaza-Schedule] section is wrong
232	The [Plaza-Schedule] – Plaza is wrong
233	The [Plaza-Schedule] – facility code is wrong
234	The [Plaza-Schedule] – plaza code is wrong
235	The [Plaza-Schedule] – direction is wrong
236	The [Plaza-Schedule] - effective date/time is wrong
237	The [Plaza-Schedule] – expiry date/time is wrong
238	The [Plaza-Schedule] – schedule-id is wrong
Ack Code Range	Ack Message
0	Success
1 - 100	File transfer error
101 – 200	TVL specific errors
201 - 300	Rate Schedule specific errors
999	Unknown Error – Contact Maintenance Group

4.5.3 Acknowledgement XSD

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="AckResultType">
    <xs:sequence>
      <xs:element name="AckCode" type="xs:int" />
      <xs:element name="AckMessage" type="xs:string" />
      <xs:element name=" FacilityCode " type="xs:string" />
      <xs:element name="Date" type="xs:string" />
      <xs:element name="FileName" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```
        <xs:element name=" InterfaceName " type="xs:string" />
        <xs:element name="TimePeriod" type="xs:string" />
    </xs:sequence>
</xs:complexType>

<xs:element name="Ack">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="AckResult" type="AckResultType" minOccurs="1"
maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>
```

5 File Transfers

5.1 Overview

File transfers are used instead of web services when bulk data or images must be transmitted.

- Bulk data can be transmitted more efficiently with a file drop architecture than with web services. Web services tend to work better when small amounts of data are needed quickly – such as when transactions are streamed to the CSC.
- Images are a special type of bulk data. Jpeg images can be embedded in XML, but the process is difficult and it has a significant amount of risk. Images can be transferred very easily, however, with file drop architecture.

5.2 TCS File Server

The TCS Vendor is responsible for providing a file server with SFTP access provided to the CSC vendor. This file server will be used for files being transferred between the TCS and the CSC Back Office. The TCS vendor is required to size and secure the server with respect to it's traffic volume specifications.

Rate Schedule and TVL Files will be pushed to the TCS File Server. Images will be pulled from the TCS File Server.

All transfers will be initiated from the CSC back office server.

5.2.1 TCS File Server Security

- Data residing on the TCS file server is classified as category 2. Category 2 data does not require encryption, but it does require authentication for access.
- SFTP is not required for this security classification, but for simplicity sake, all file transfers to and from the CSC back office is accomplished with SFTP. FTP is not allowed.
- Authentication and encryption will be accomplished with SSH v2 public key / private key pairs.
 - The key will be generated by the CSC vendor using the Linux “ssh-keygen -t dsa” command. SSH-keygen an SSLv2 public key / private key pair.
 - The default key is a Rivest, Shamir, and Adleman (RSA) key. The “-t dsa” specifies a Digital Signature Algorithm (DSA) key. The DSA key will be used exclusively, as it is more secure.
 - The private key is kept securely on the CSC back office server.
 - The CSC public key will be given to the TCS vendors to be placed in the authorized keys file. It is normally kept at ~home/.ssh/known_hosts on Linux.
 - Placing the CSC public key on the TCS file server establishes a trust relationship between the servers. This trust does not prompt for username/password and encrypts all transfers using the DSA key.
 - DSA keys are exactly 1024 bits as specified in Federal Information Processing Standard (FIPS) 186-2.
- The public key / private keys should be replaced annually in accordance with password management policies.

5.3 Transponder Validation List (TVL)

5.3.1 Overview

The Transponder Validation List (TVL) is created by the CSC and transmitted to the TCS File Server for processing. It contains the transponder status data for all transponders associated with an account maintained at the CSC.

The TVL will be pushed nightly from the CSC to the TCS File Server.

5.3.2 Transfer Protocol

Secure File Transfer Protocol (SFTP) will be used for all TVL transfers between the CSC Back Office and the TCS File Server.

5.3.3 File Archival

TVL files should be retained on the TCS File Server for a period of 30 days. This retention period is not required by the CSC interface. It is suggested in order to support data analysis in the event there are issues with the data.

The TCS Vendor is responsible for purging TVL files that are older than 30 days.

5.3.4 Compression

TVL files transmitted to the TCS File Server will be compressed using a standard Linux zip utility.

The specific version used is:

- Linux zip 2.31 (Mar 8, 2005).

It has been successfully tested with

- Linux zip 2.31 (Mar 8 2005).
- Windows zip 2.1 (Apr 27, 1996)

Zip files will be pre-tested before implementation to ensure files zipped by the CSC Back Office can be successfully unzipped by TCS Vendors.

The file extension will be converted from file-name.file-type to file-name_file-type.ZIP.

Example:

For a Full TVL file named, 20060619020005_Full_TV.LXML, the zipped file name will be 20060619020005_Full_TV.LXML.ZIP.

5.3.5 File Types

5.3.5.1 Full TVL File

The full TVL file contains all transponder status data for all transponders maintained at the CSC. Since the CSC contains transponders that are expired or invalid, the TVL will necessarily include transponders with expired or invalid status codes.

The full TVL file will be transmitted to the TCS File Server daily.

5.3.5.2 Incremental TVL File

The incremental TVL file contains only those transponders that have changed since the last full or incremental TVL file was generated.

The incremental TVL file will be transmitted to the TCS File Server hourly.

5.3.6 Naming Convention

[FILE_DATE_TIME]_Full_TVL.XML Full TVL File
 [FILE_DATE_TIME]_Incr_TVL.XML Incremental TVL File

Example: For a Full TVL File created on June 19, 2006 at 02:00:05 the name of the file will be:

20060619020005_Full_TVL.XML

The zipped version of this file will be:

20060619020005_Full_TVL_XML.ZIP

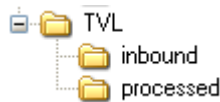
5.3.7 Acknowledgement

After receiving the file, TCS will send an Ack/Nak to the CSC by calling CSC’s web service. See the Acknowledgement Section.

5.3.8 Directory Structure

The first level of the directory structure will be “TVL”. TVL files will be pushed to the TVL/inbound directory. As TVL files are processed, they will be moved to the TVL/processed directory. The TCS Vendor is free to create additional directories as needed, but these 2 directories must exist.

Here is an example of the directory structure:



5.3.9 Data Elements

TagName	Required?	Data Type	Notes
TransponderValidationList	Required	ComplexType	
Header			
FileGenerateTime	Required	dateTime	
TotalRecordCount	Required	long	Total count of records in the file AccountVehicle minOccurs="1" maxOccurs="unbounded"
Body			
AccountVehicle minOccurs="1" maxOccurs="unbounded"			
CSCClassification	Optional	string	Vehicle classification as identified by TVL.

			Domain: VehicleClassTypes
TransponderID	Required	string	Transponder Serial Number
TransponderAgencyCode	Required	string	Domain: AgencyCodes
TransponderProtocol	Optional	string	Domain: TransponderProtocols
TransponderStatusCode	Required	string	Domain: TransponderStatusCodes

5.3.10 File Example

```
<?xml version="1.0"?>
  <TransponderValidationList>
    <Header>
      < FileGenerateTime>2010-07-27 00:00:00</ FileGenerateTime>
      < TotalRecordCount>2</ TotalRecordCount>
    </Header>
    <Body>
      <AccountVehicle>
        <CSCClassification> 2 </CSCClassification>
        <TransponderID> 1234567890 </TransponderID>
        <TransponderAgencyCode> 1 </TransponderAgencyCode>
        <TransponderProtocol> SeGO </TransponderProtocol>
        <TransponderStatusCode> 1 </TransponderStatusCode>
      </AccountVehicle>
      <AccountVehicle>
        <CSCClassification> 2 </CSCClassification>
        <TransponderID> 2345678901 </TransponderID>
        <TransponderAgencyCode> 1 </TransponderAgencyCode>
        <TransponderProtocol> SeGO </TransponderProtocol>
        <TransponderStatusCode> 1 </TransponderStatusCode>
      </AccountVehicle>
    </Body>
  </TransponderValidationList>
```

5.4 Rate Schedule

5.4.1 Overview

The Rate Schedule is maintained by the CSC and transmitted to the TCS File Server for processing. It contains the toll rates in effect for each vehicle class.

5.4.2 Frequency

The Rate Schedule file will be pushed from the CSC Back Office to the TCS File Server as needed. A notification email will be sent out to a pre-defined TCS maintenance group when a rate schedule file is pushed.

The TCS vendors should look for new rate schedule files at least hourly. The CSC vendor will try to ensure that the rate schedules are posted at least an hour before they go into effect, but in some cases, especially in emergency situations, the rate schedules might be posted with less notice. In these cases, the CSC vendor will notify the TCS vendors so that they can ensure that the proper rates are posted.

5.4.3 Transfer Protocol

Secure File Transfer Protocol (SFTP) will be used for all Rate Schedule transfers between the CSC Back Office and the TCS File Server.

5.4.4 File Archival

Rate Schedule files will be retained on the TCS File Server for a period of 30 days.

The TCS Vendor is responsible for purging Rate Schedule files that are older than 30 days.

5.4.5 File Type

Rate Schedule file is an ASCII text file.

5.4.6 Naming Convention

[FILE_DATE_TIME]_Rate_Schedule.txt

Example: For a Rate Schedule File created on June 19, 2006 at 02:00:05 the name of the file will be:

20060619020005_Rate_Schedule.txt

5.4.7 Acknowledgement

After receiving rate schedule file, TCS will send an Ack/Nak to the CSC by calling CSC's web service. See the Acknowledgement Section.

5.4.8 Directory Structure

The first level of the directory structure will be "rate_schedule". Rate Schedule files will be pushed to the rate_schedule/inbound directory. As rate schedule files are processed, they will be moved to the rate_schedule/processed directory. The TCS Vendor is free to create additional directories as needed, but these 2 directories must exist.

Here is an example of the directory structure:



5.4.9 File Format

Rate Schedule file is modelled after a Microsoft Windows' INI file.

-Section Heading – Section Heading is contained in square brackets (“[]”).

-Parameter – Parameters are identified by the first keyword on a line, followed by an equal sign (“=”), then the parameter value.

-Comment – Comment begin with a semi-colon (“;”). Any text following a semi-colon on a line is ignored.

Section Heading and parameter are not case-sensitive.

A Rate Schedule file contains three sections:

- Fare structure definition: “[FareStructure]” section;
- 7-day, 24-hours schedule definitions: “[Schedule-Fare]” section;
- Plaza to Schedule mappings: “Plaza-Schedule]” section;

A Rate Schedule file may contain multiple fare structures, schedule definitions, and plaza-to-schedules mappings. Each schedule definition section heading begins a new rate schedule and spans an entire 7-day, 24-hour period. A simple rate schedule for the entire period may be represented as the example below:

Day Of Week	Hour of the Day																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
S	\$1.00																							
M	\$1.00			\$2.50			\$1.00						\$2.50			\$1.00								
T	\$1.00																							
W	\$1.00																							
T	\$1.00																							
F	\$1.00			\$2.50			\$1.00						\$2.50			\$1.00								
S	\$1.00																							

In text format, the above schedule is represented as:

Start		End		Fare
Day	Time	Day	Time	
SUN	00:00	MON	06:00	\$1.00
MON	06:00	MON	09:00	\$2.50
MON	09:00	MON	16:00	\$1.00
MON	16:00	MON	19:00	\$2.50
MON	19:00	FRI	06:00	\$1.00
FRI	06:00	FRI	09:00	\$2.50
FRI	09:00	FRI	16:00	\$1.00
FRI	16:00	FRI	19:00	\$2.50
FRI	19:00	SUN	00:00	\$1.00

Notice, Sunday midnight is Day SUN, 00:00. As an end point, the Saturday/Sunday midnight boundary is specified as Day SUN, 00:00. The schedule must begin on Day SUN, 00:00 and end on Day SUN, 00:00.

5.4.10 Record Format

5.4.10.1 [FareStructure] Section

The FareStructure section specifies the fare due by vehicle class and payment type.

This section contains the following parameters:

```
[FareStructure]           ; Section heading
FareID=<fare-id>         ; numeric identifier for this fare
                          ; structure ranging from 1 to 65535.
Fare=<class>, <payment-type>, <fare-amount> [, <per-axle-fare>]
                          ; where:
                          ;   class = vehicle class
                          ;   payment type = 1 = cash
                          ;           2 = AVI (revenue)
                          ;           8 = credit AVI
                          ;   fare amount = minimum fare due for
                          ;   class (implied decimal)
                          ;   e.g., 100 = $1.00
                          ;   per-axle-fare = per axle charge if
                          ;   fare based upon axle count
                          ;   (default 0, optional)
                          ; Multiple Fare keywords are allowed
                          ; per section
```

Business Rules:

- Only one FareID is allowed per section;
- The FareID should be the first entry in the section;
- Any vehicle class-payment type combination omitted from this section is assumed to have a value of zero;
- A new fare schedule definition begins with another [FareStructure] section heading.

5.4.10.2 [Schedule-Fare] Section

The Schedule-Fare Section maps a set of fare definitions to a 7-day, 24-hour/day period. The association to a given plaza is specified in the Plaza-Schedule Section described later in this document.

This section contains the following parameters:

```
[Schedule-Fare]           ; Section heading
ScheduleID=<schedule-id> ; numeric identifier for the schedule
                          ; ranging from 1 to 655351. Schedule ID 0
                          ; is reserved for legacy rate schedules
                          ; that do not support variable pricing
Period=<start-day>,       ; starting day:
                          ; SUN = Sunday
```

```

; MON = Monday
; TUE = Tuesday
; WED = Wednesday
; THU = Thursday
; FRI = Friday
; SAT = Saturday
<start-time>, ; start time of price segment
; format HH:MM (24-hr)
<end-day>, ; ending day (same abbreviations as
; start-day)
<end-time>, ; end time of price segment
; format HH:MM (24-hr)
<fare-id> ; applicable fare ID, the fare ID must
; defined in a previous FareStructure
; Section.
; Multiple Period keywords are allowed
; per section.

```

Business Rules:

- Only one ScheduleID is allowed per section;
- The ScheduleID should be the first entry in the section;
- Each "Period=" line contains an effective time range and an associated fare structure.
- The time is specified as HOUR:MINUTE. The implication is :00 seconds.
- The entire 7-day period is specified as starting "SUN, 00:00" and ending "SUN, 00:00". If a fare schedule is omitted from a specified period, the fare amount is set to zero.
- A new schedule begins with another [Schedule-Fare] section heading.

5.4.10.3 [Plaza-Schedule] Section

The Plaza-Schedule Section specifies active schedule for a given plaza. This section contains the following parameters:

```

[Plaza-Schedule] ; Section heading
Plaza=<facility code>, ; 4-char facility ID
<plaza code>, ; 5-char plaza ID
<direction>, ; travel direction (N, S, E, W)
<effective date/time>, ; start date of rate
; format MM/DD/YYYY HH:MM (24-hr)
<expiry date/time>, ; date rate expires
; format MM/DD/YYYY HH:MM (24-hr)
; 99/99/9999 23:59 = no expiration
<schedule-id> ; Schedule ID

```

Business Rules:

- Multiple schedules may be specified for a given plaza. This file will be processed from start to finish by the lane. The last schedule in the file specified for a given plaza with an applicable time frame will be used by the lane. This allows an override schedule to be appended to the section which would preempt a prior schedule. For example, to override a schedule for July 4th, 2006 a portion of the Lane Variable Rate File might be:

```

[Plaza-Schedule] ; Section heading

```

; The following line is the "default schedule"
Plaza=ABC, MLP1, N, 01/01/2004 00:00, 99/99/9999 23:59, 1
; The following line would override the schedule for 1 day
Plaza=ABC, MLP1, N, 07/04/2006 00:00, 07/05/2006 00:00, 2

In the above example, Schedule 1 is active from January 1st, 2004 with no end time specified. However, at midnight July 4th, 2006, Schedule 2 becomes active until midnight July 5th, 2006. At midnight July 5th, 2006, Schedule 2 terminates and Schedule 1 becomes active once again.

-The "Plaza=" entry is repeated multiple times in this section to specify the schedule applicable to multiple plazas.

A sample Rate Schedule file is attached in the Appendix.

5.5 Images

5.5.1 Overview

The TCS collects video images of vehicles on the toll road. These images are used for photo-enforced tolls. Detailed information about the images is recorded in the toll transaction data exchange.

Images will be pulled by the CSC from the TCS File Server.

5.5.2 Transfer Protocol

Secure File Transfer Protocol (SFTP) will be used to pull images from the TCS File Server to the CSC Back Office.

5.5.3 File Archival

Images must be retained on the TCS File Server for at least 90 days. The TCS Vendor is responsible for purging images that are older than 90 days.

5.5.4 Compression

Images will not be compressed. Jpeg images do not benefit from compression.

5.5.5 Naming Convention

Images will be named in a unique manner by the TCS Vendor. The image names will be specified in the toll transaction.

In order to ensure image names are unique, the following data elements will be specified:

YYYYMMDDHH24MISS_Facility_Plaza_Lane#_Sequence#_Image#_ImageType.JPG

- YYYYMMDD – Transaction Year/Month/Day
- HH24MISS – Transaction Hour/Minute/Second
- Facility – Facility Code
- Plaza – Plaza Code
- LaneNumber – Lane Number
- Sequence# – Sequential number used to unique identify a transaction
- Image# – Used to uniquely identify an image.

- ImageType – FSI=Front Shot Image, BSI=Back Shot Image, ROI=Region of Interest, EFI=Entry Front Shot Image (HOT Lanes), EBI=Entry Back Shot Image (Hot Lanes)

Examples:

20100727113029_520_520EB_01_123456_01_FSI.JPG
20100727113029_520_520EB_01_123456_01_ROI.JPG
20100727113029_520_520EB_01_123456_02_BSI.JPG
20100727113029_520_520EB_01_123456_02_ROI.JPG

5.5.6 Directory Structure

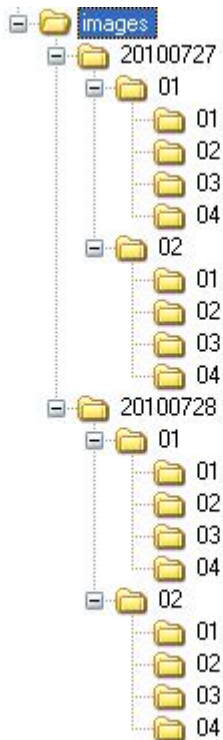
In order avoid performance issues, the directory structure will be formed by using the transaction date along with the plaza number and lane number.

For open road tolling, the toll zone number will be used instead of the plaza number.

The first level of the directory structure will be “images” with the YYYYMMDD/Plaza#/Lane# structure following.

This directory structure will be used for the ImageLocation data element. i.e. images/20100727/01/01.

Here is an example of the directory structure for 4 lanes (01, 02, 03, 04) in 2 plazas (01, 02) over 2 days (07/27/2010 and 07/28/2010):



5.5.7 Regions of Interest (ROI)

The TCS Vendor is responsible for identifying regions of interest and performing OCR.

5.5.7.1 Separate Images for ROI

Some TCSs identify the ROI by X and Y coordinates. Other TCSs identify the ROI by cropping the regions of interest and providing separate images.

In order to maintain consistency across all TCSs, this interface requires TCSs to provide separate images for regions of interest. This requires vendors that usually provide X and Y coordinates to crop the images instead of passing X and Y coordinates.

5.5.7.2 Asynchronous OCR Process

Some TCSs perform OCR as images are collected at the roadside. These systems can easily provide OCR results when toll transactions are submitted.

Other TCSs perform OCR in batch. These systems cannot easily provide OCR results when toll transactions are submitted.

In order to have consistency across all TCSs, this interface requires roadside systems to provide OCR results in the TollTransaction web service. This requires the vendors that perform OCR in batch to match toll transactions to OCR results after the images have been processed by the OCR software. This will necessarily delay toll transactions until OCR processing is complete. The delay is expected to be between 1-2 hours due to the normal cycles of OCR processing. If the delay exceeds 4 hours, the TCS Vendor should notify the CSC Vendor.

6 Operations

6.1 *Scheduled Maintenance*

The CSC Vendor's maintenance window for the CSC Back Office is defined to be the 3rd Saturday of each month beginning on that Saturday at 7pm and lasting until the next morning (Sunday) at 7am. All routine preventative maintenance, patches and upgrades will be performed during this maintenance window. During these times, the interfaces specified in this ICD may become unavailable for all or a period of this defined window.

Scheduled maintenance is performed in accordance with a mature change management process. This process involves several steps including *approval* by the CSC project team (WSDOT) and *notification* to all impacted parties. Notification will be sent prior to each scheduled maintenance to provide more details about the expected impacts to the various interfaces.

In the event additional maintenance windows are necessary the CSC Vendor will coordinate with interested parties to secure an appropriate time for a particular outage.

Scheduled maintenance (outages) is required for patches and upgrades. The disaster recovery site is an active-passive configuration; data is replicated from the primary site to the D/R site, to facilitate preparedness for a failover in the event of a disaster. Scheduled maintenance is not considered a disaster event.

The expectation at notification is maintenance will end at the appointed time listed in the notice. A notice of maintenance completion will only be sent to appropriate parties in the event the maintenance uncovered an issue that would require more time than the maintenance window allows.

Unscheduled Maintenance

Unscheduled maintenance is generally used to respond to problems that may arise and need to be corrected before the next scheduled maintenance window.

For this time of maintenance activity normally occurs in the evenings and is always preceded by notifications of the coming outage.

6.2 *Unscheduled Outages*

In the event the system is unavailable, toll transactions must be queued at the TCS so they may be transmitted when the system becomes available.

If the interface to the CSC is unavailable and there is no scheduled maintenance, the TCS Vendor should notify the CSC Vendor for more information about the outage. When the interface becomes available, the CSC Help Desk will notify affected parties.

The Service Level Agreement (SLA) of the CSC states that the interface may be down for up to 24 hours. The TCS vendors must have sufficient storage capacity to queue 24 hours worth of transactions.

6.3 Change Management

Any changes to this ICD and the interfaces will go through a rigorous change management process. This will include contacting and engaging all partners in the process, analyzing impacts, and coordinating implementation.

6.4 Troubleshooting

All web service messages are retained in the CSC database in their unaltered / raw format. The raw messages may be accessed to support auditing or data research activities.

The raw messages are purged from the CSC database after 90 days.

6.5 Disaster Recovery / Business Continuity

The state wide CSC program includes a primary data center and a disaster recovery data center. In the event the primary data center is unavailable, processing may be diverted to the disaster recovery data center. This will necessarily require the TCS vendors to redirect TollTransactions during the diversion. The method and details of the redirection as well as all other interfaces between the CSC and the TCS will be defined in the Business Continuity Plan. Sections pertaining to this interface will be extracted and included in this ICD. For long term network outages between the TCS and the CSC, a Laptop computer with the required software to simulate the CSC interface will be on hand locally to allow transactions and files to be downloaded and uploaded to the TCS if required.

6.6 Contact Information

The following information is provided for support. The help desk should be contacted first. If the help desk is unavailable or does not provide sufficient support, the issue should be escalated to the program manager. If the program manager is unavailable or does not provide sufficient support, the issue should be escalated to the program owner.

Details	Program Owner	Program Manager	Help Desk
Name	Lucinda Broussard	Will Hanks	ETCC Help Desk
Title	CSC Toll Operations Manager, WSDOT	CSC Project Manager Operations, ETCC	
Address	401 2 nd Ave South Seattle, WA 98104	401 2 nd Ave South Seattle, WA 98104	1705 North Plano Rd Richardson, TX 75081
Phone Number	(253) 534-4699	(206) 387-2492	(214) 451-4860
E-mail Address	broussl@wsdot.wa.gov	Will.Hanks@etcc.com	etcctechsupport@etcc.com
Notes			Days and hours of support will be defined prior to go-live.

Appendix A Toll Transaction Web Service

A.1 XSD

```
<xsd:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="TollTransaction">
    <xsd:complexType>
      <xsd:element name="FacilityCode" type="FacilityCodeType"
        minOccurs="1" />
      <xsd:element name="PlazaCode" type="PlazaCodeType"
        minOccurs="1" />
      <xsd:element name="LaneNumber" type="LaneNumberType"
        minOccurs="1" />
      <xsd:element name="TransactionDateTime" type="xsd:dateTime"
        minOccurs="1" />
      <xsd:element name="EntryFacilityCode" type="FacilityCodeType"
        minOccurs="0" />
      <xsd:element name="EntryPlazaCode" type="PlazaCodeType"
        minOccurs="0" />
      <xsd:element name="EntryLaneNumber" type="LaneNumberType"
        minOccurs="0" />
      <xsd:element name="EntryTransactionDateTime" type="xsd:dateTime"
        minOccurs="0" />
      <xsd:element name="SequenceNumber" type="SequenceNumberType"
        minOccurs="1" />
      <xsd:element name="TransactionType" type="TransactionTypeType"
        minOccurs="1" />
      <xsd:element name="AVCClassification" type="VehicleClassType"
        minOccurs="1" />
      <xsd:element name="CSCClassification" type="VehicleClassType"
        minOccurs="1" />
      <xsd:element name="Transponder" type="TransponderType"
        minOccurs="0" />
      <xsd:element name="ImageList" type="ImageListType"
        minOccurs="0" />
      <xsd:element name="FareAmount" type="FareAmountType"
        minOccurs="0" />
      <xsd:element name="VehicleSpeed" type="VehicleSpeedType"
        minOccurs="0" />
      <xsd:element name="UnusualOccurrences" type="UnusualOccurrences"
        minOccurs="0" />
      <xsd:element name="TCSStatus" type="TCSStatusType"
        minOccurs="0" />
      <xsd:element name="ReverseDirectionFlag"
        type="xsd:boolean"
        minOccurs="0" />
      <xsd:element name="LaneMode" type="LaneModeType"
        minOccurs="1" />
      <xsd:element name="AxleCount" type="AxleCountType"
        minOccurs="1" />
    </xsd:complexType>
  </xsd:element>

  <xsd:simpleType name="FacilityCodeType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="TNB"/>
      <xsd:enumeration value="SR167"/>
      <xsd:enumeration value="SR520"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>
```

```
</xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="PlazaCodeType">
  <xsd:restriction base="xsd:string">
    <xsd:length value="5">
    </xsd:restriction>
  </xsd:simpleType>

<xsd:simpleType name="LaneNumberType">
  <xsd:restriction base="xsd:int">
    <minInclusive value="1"/>
    <maxInclusive value="99"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="SequenceNumberType">
  <xsd:restriction base="xsd:long">
    <minInclusive value="0"/>
    <maxInclusive value="9999999999"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="TransactionTypeType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="1"/>
    <xsd:enumeration value="2"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="VehicleClassType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="1"/>
    <xsd:enumeration value="2"/>
    <xsd:enumeration value="3"/>
    <xsd:enumeration value="4"/>
    <xsd:enumeration value="5"/>
    <xsd:enumeration value="6"/>
    <xsd:enumeration value="7"/>
    <xsd:enumeration value="8"/>
    <xsd:enumeration value="9"/>
    <xsd:enumeration value="10"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="TransponderIDType">
  <xsd:restriction base="xsd:string">
    <xsd:length value="10">
    <xsd:pattern value="\d{10}"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="AgencyCodeType">
  <xsd:restriction base="xsd:int">
    <minInclusive value="0"/>
    <maxInclusive value="99"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="TransponderStatusType">
```

```
<xsd:restriction base="xsd:string">
  <xsd:enumeration value="0"/>
  <xsd:enumeration value="1"/>
  <xsd:enumeration value="2"/>
  <xsd:enumeration value="3"/>
  <xsd:enumeration value="4"/>
  <xsd:enumeration value="5"/>
  <xsd:enumeration value="6"/>
  <xsd:enumeration value="7"/>
  <xsd:enumeration value="8"/>
</xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="DeclarationStateType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="0"/>
    <xsd:enumeration value="1"/>
    <xsd:enumeration value="2"/>
    <xsd:enumeration value="3"/>
    <xsd:enumeration value="4"/>
    <xsd:enumeration value="5"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="FareAmountType">
  <xsd:restriction base="xsd:decimal">
    <totalDigits value="5"/>
    <fractionDigits value="2"/>
    <minInclusive value="0"/>
    <maxInclusive value="100"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="VehicleSpeedType">
  <xsd:restriction base="int">
    <totalDigits value="2"/>
    <minInclusive value="0"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="TCSStatusType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="1"/>
    <xsd:enumeration value="2"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="LaneModeType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="0"/>
    <xsd:enumeration value="1"/>
    <xsd:enumeration value="2"/>
    <xsd:enumeration value="3"/>
    <xsd:enumeration value="4"/>
    <xsd:enumeration value="5"/>
    <xsd:enumeration value="6"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="AxleCountType">
```

```
<xsd:restriction base="int">
  <totalDigits value="2"/>
  <minInclusive value="0"/>
  <maxInclusive value="15"/>
</xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="ImageLocationType">
  <xsd:restriction base="xsd:string">
    <maxLength value="150"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="ImageNameType">
  <xsd:restriction base="xsd:string">
    <maxLength value="60"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="ImageTypeType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="ROI"/>
    <xsd:enumeration value="FSI"/>
    <xsd:enumeration value="BSI"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="PlateNumberType">
  <xsd:restriction base="xsd:string">
    <maxLength value="15"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="PlateStateType">
  <xsd:restriction base="xsd:string">
    <maxLength value="3"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="PlateTypeType">
  <xsd:restriction base="xsd:string">
    <maxLength value="3"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="PercentType">
  <xsd:restriction base="xsd:int">
    <minInclusive value="0"/>
    <maxInclusive value="100"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="TransponderType">
  <xsd:element name="TransponderID"
    type="TransponderIDType" />
  <xsd:element name="AgencyCode"
    type="AgencyCodeType" />
  <xsd:element name="Status"
    type="TransponderStatusType"/>
  <xsd:element name="DeclarationState"
```

```

        type="DeclarationStateType"
        minOccurs="0"/>
    <xsd:element name="LowBattery"
        type="xsd:boolean"
        minOccurs="0"/>
</xsd:complexType>

<xsd:complexType name="ImageListType">
    <xsd:element name="ImageLocation" type="ImageLocationType"/>
    <xsd:element name="SelectedImageName" type="ImageNameType"/>
    <xsd:element name="SelectedROIImageName" type="ImageNameType"/>
    <xsd:element name="Image" minOccurs="0" maxOccurs="unbounded">
        <xsd:sequence>
            <xsd:element name="ImageName" type="ImageNameType"/>
            <xsd:element name="ImageType" type="ImageTypeType"/>
            <xsd:element name="OCRPlateNumber" type="PlateNumberType"
                minOccurs="0" />
            <xsd:element name="OCRPlateState" type="PlateStateType"
                minOccurs="0" />
            <xsd:element name="OCRPlateType" type="PlateTypeType"
                minOccurs="0" />
            <del><xsd:element name="OCRConfidence" type="PercentType"
                minOccurs="0" /></del>
            <xsd:element name="OCRPlateNumberConfidence" type="PercentType"
                minOccurs="0" />
            <xsd:element name="OCRPlateStateConfidence" type="PercentType"
                minOccurs="0" />
            <xsd:element name="OCRPlateTypeConfidence" type="PercentType"
                minOccurs="0" />
        </xsd:sequence>
    </xsd:element>
</xsd:complexType>

<xsd:complexType name="TollTransactionResponse">
    <xsd:element name="ResponseCode" type="xsd:string" />
    <xsd:element name="ResponseMessage" type="xsd:string" />
    <xsd:element name="TransactionSequenceNumber"
        type="xsd:long"/>
    <xsd:element name="TransactionId" type="xsd:long"/>
</xsd:complexType>
</xsd:schema>

```

A.2 WSDL

```
<wsdl:definitions
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.example.org/TollTransaction/"
  xmlns:types="urn:TollTransactionDataTypes"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  name="TollTransaction"
  targetNamespace="http://www.example.org/TollTransaction/">

  <wsdl:import location="TollTransaction.xsd"
    namespace="urn:TollTransactionDataTypes" />

  <wsdl:message name="TollTransaction_IN">
    <wsdl:part name="TollTransaction" type="types:TollTransaction" />
  </wsdl:message>

  <wsdl:message name="TollTransaction_OUT">
    <wsdl:part name="response" type="types:TollTransactionResponse" />
  </wsdl:message>

  <wsdl:portType name="TollTransactionInterface">
    <wsdl:operation name="sendTransactionData">
      <wsdl:input message="tns:TollTransaction_IN" />
      <wsdl:output message="tns:TollTransaction_OUT" />
    </wsdl:operation>
  </wsdl:portType>

  <wsdl:binding name="TollTransactionSOAP"
    type="tns:TollTransactionInterface">
    <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="sendTransactionData">
      <soap:operation soapAction =
        "http://www.example.org/TollTransaction/sendTransactionData"
        <wsdl:input>
          <soap:body use="literal"
            namespace="http://www.example.org/TollTransaction/"
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal"
            namespace="http://www.example.org/TollTransaction/"
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
  <wsdl:service name="TollTransaction">
    <wsdl:port binding = "tns:TollTransactionSOAP"
      name="TollTransactionSOAP">
      <soap:address location="http://www.example.org/" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Appendix B Daily Reconciliation Web Service

B.1 XSD

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="DailyReconciliation">
    <xsd:complexType>
      <xsd:element name="FromDate" type="xsd:DateTime" minOccurs="1" />
      <xsd:element name="ToDate" type="xsd:DateTime" minOccurs="1" />
      <xsd:element name="ArrayList" type="ArrayListType" minOccurs="1" />
    </xsd:complexType>
  </xsd:element>

  <xsd:simpleType name="FacilityCodeType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="TNB"/>
      <xsd:enumeration value="SR167"/>
      <xsd:enumeration value="SR520"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="PlazaCodeType">
    <xsd:restriction base="xsd:string">
      <xsd:length value="5"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="LaneNumberType">
    <xsd:restriction base="xsd:int">
      <minInclusive value="1"/>
      <maxInclusive value="99"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="TransactionTypeType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="1"/>
      <xsd:enumeration value="2"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="VehicleClassType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="1"/>
      <xsd:enumeration value="2"/>
      <xsd:enumeration value="3"/>
      <xsd:enumeration value="4"/>
      <xsd:enumeration value="5"/>
      <xsd:enumeration value="6"/>
      <xsd:enumeration value="7"/>
      <xsd:enumeration value="8"/>
      <xsd:enumeration value="9"/>
      <xsd:enumeration value="10"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:complexType name="ArrayType">
    <xsd:element name="ArrayElement" minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="FacilityCode" type="FacilityCodeType" />
    </xsd:element>
  </xsd:complexType>
</xsd:schema>
```

```
<xsd:element name="PlazaCode" type="PlazaCodeType" />
<xsd:element name="LaneNumber" type="LaneNumberType" />
<xsd:element name="TransactionType" type="TransactionTypeType" />
<xsd:element name="AVClassification" type="VehicleClassType" />
<xsd:element name="TransactionCount" type="xsd:long" />
</xsd:element>
</xsd:complexType>

<xsd:complexType name="CommonResponse">
  <xsd:element name="ResponseCode" type="xsd:string" />
  <xsd:element name="ResponseMessage" type="xsd:string" />
</xsd:complexType>

</xsd:schema>
```

B.2 WSDL

```
<wsdl:definitions
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.example.org/DailyReconciliation/"
  xmlns:types="urn:DailyReconciliationDataTypes"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  name=" DailyReconciliation"
  targetNamespace="http://www.example.org/DailyReconciliation">
  <wsdl:import location=" DailyReconciliation.xsd"
    namespace="urn:DailyReconciliationDataTypes " />
  <wsdl:message name=" dailyReconciliation_IN">
    <wsdl:part name=" DailyReconciliation"
      type="types: DailyReconciliation" />
  </wsdl:message>
  <wsdl:message name=" dailyReconciliation_OUT">
    <wsdl:part name="DailyReconciliationResponse"
      type="types:CommonResponse" />
  </wsdl:message>
  <wsdl:portType name="DailyReconciliationInterface">
    <wsdl:operation name="sendDailyReconciliationReport">
      <wsdl:input message="tns:dailyReconciliation_IN" />
      <wsdl:output message="tns:dailyReconciliation_OUT" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="DailyReconciliationSOAP"
    type="tns:DailyReconciliationInterface">
    <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="sendDailyReconciliationReport">
      <soap:operation
        soapAction="http://www.example.org/DailyReconciliation/sendDailyReconciliationReport"/>
      <wsdl:input>
        <soap:body use="literal"
          namespace="http://www.example.org/DailyReconciliation/"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"
          namespace="http://www.example.org/DailyReconciliation/"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="DailyReconciliation">
    <wsdl:port binding="tns:DailyReconciliationSOAP"
      name="DailyReconciliationSOAP">
      <soap:address location="http://www.example.org/" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Appendix C AcknowledgeFile

C.1 XSD

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  attributeFormDefault="qualified" elementFormDefault="qualified"
  targetNamespace="http://types.rim.biz.core.host.etcc.com">
  <xsd:complexType name="AcknowledgeFile">
    <xsd:sequence>
      <xsd:element name="AckCode" type="xsd:int" />
      <xsd:element minOccurs="0" name="AckMessage" nillable="true" type="xsd:string" />
      <xsd:element name="FacilityCodes" type="xsd:string" />
      <xsd:element minOccurs="0" name="Date" nillable="true" type="xsd:string" />
      <xsd:element name="FileName" type="xsd:string" />
      <xsd:element name="InterfaceName" type="xsd:string" />
      <xsd:element minOccurs="0" name="TimePeriod" nillable="true" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="CommonResponse">
    <xsd:sequence>
      <xsd:element name="responseCode" type="xsd:string" />
      <xsd:element name="responseMessage" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

C.2 WSDL

```
<?xml version="1.0" encoding="UTF-8" ?>
<definitions targetNamespace="http://rim.biz.core.host.etcc.com"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:ns1="http://types.rim.biz.core.host.etcc.com"
  xmlns:soapenc12="http://www.w3.org/2003/05/soap-encoding"
  xmlns:tns="http://rim.biz.core.host.etcc.com"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soapenc11="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <message name="sendAcknowledgmentRequest">
    <part name="in0" type="ns1:AcknowledgeFile" />
  </message>
  <message name="sendAcknowledgmentResponse">
    <part name="out" type="ns1:CommonResponse" />
  </message>
```

```
</message>
<portType name="AcknowledgeFileInterfacePortType">
<operation name="sendAcknowledgment">
  <input name="sendAcknowledgmentRequest"
    message="tns:sendAcknowledgmentRequest" />
  <output name="sendAcknowledgmentResponse"
    message="tns:sendAcknowledgmentResponse" />
</operation>
</portType>
<binding name="AcknowledgeFileInterfaceHttpBinding"
  type="tns:AcknowledgeFileInterfacePortType">
  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
<operation name="sendAcknowledgment">
  <wsdlsoap:operation soapAction="" />
<input name="sendAcknowledgmentRequest">
  <wsdlsoap:body use="literal" namespace="http://rim.biz.core.host.etc.com" />
</input>
<output name="sendAcknowledgmentResponse">
  <wsdlsoap:body use="literal" namespace="http://rim.biz.core.host.etc.com" />
</output>
</operation>
</binding>
<service name="AcknowledgeFileInterface">
<port name="AcknowledgeFileInterfaceHttpPort"
  binding="tns:AcknowledgeFileInterfaceHttpBinding">
  <wsdlsoap:address
    location="http://localhost:8888/host_ui/services/AcknowledgeFileInterface" />
</port>
</service>
</definitions>
```

Appendix D TransponderValidationList

D.1 XSD

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="TransponderValidationList">
    <xsd:element name="Header" type="HeaderType" />
    <xsd:element name="Body" type="BodyType" />
  </xsd:element>

  <xsd:complexType name="HeaderType">
    <xsd:element name="FileGenerateTime" type="xsd:dateTime" />
    <xsd:element name="TotalRecordCount" type="xsd:long" />
  </xsd:complexType>

  <xsd:complexType name="BodyType">
    <xsd:element name="AccountVehicle"
      minOccurs="0" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:element name="CSCClassification"
          type="CSCClassificationType"/>
        <xsd:element name="TransponderID"
          type="TransponderIDType" />
        <xsd:element name="TransponderAgencyCode"
          type="AgencyCodeType" />
        <xsd:element name="TransponderProtocol"
          type="TransponderProtocolType"/>
        <xsd:element name="TransponderStatusCode"
          type="TransponderStatusType"/>
      </xsd:complexType>
    </xsd:element>
  </xsd:complexType>

  <xsd:simpleType name="CSCClassificationType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="0"/>
      <xsd:enumeration value="1"/>
      <xsd:enumeration value="2"/>
      <xsd:enumeration value="3"/>
      <xsd:enumeration value="4"/>
      <xsd:enumeration value="5"/>
      <xsd:enumeration value="6"/>
      <xsd:enumeration value="7"/>
      <xsd:enumeration value="8"/>
      <xsd:enumeration value="9"/>
      <xsd:enumeration value="10"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="TransponderIDType">
    <xsd:restriction base="xsd:string">
      <xsd:length value="10">
      <xsd:pattern value="\d{10}"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="AgencyCodeType">
```

```
<xsd:restriction base="xsd:int">
  <minInclusive value="0"/>
  <maxInclusive value="99"/>
</xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="ProtocolType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="SeGo"/>
    <xsd:enumeration value="ASTM"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="TransponderStatusType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="0"/>
    <xsd:enumeration value="1"/>
    <xsd:enumeration value="2"/>
    <xsd:enumeration value="3"/>
    <xsd:enumeration value="4"/>
    <xsd:enumeration value="5"/>
    <xsd:enumeration value="6"/>
    <xsd:enumeration value="7"/>
    <xsd:enumeration value="8"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>
```

Appendix E Rate Schedule Sample File

A sample Lane Variable Rate Schedule File follows.

; Created 10/01/2004 08:00

```
[FareStructure]
FareID=1
Fare=2, 1, 50           ; Class 2, cash (type 1)
Fare=3, 1, 75           ; Class 3
Fare=4, 1, 100          ; Class 4
Fare=5, 1, 125          ; Class 5
Fare=6, 1, 150          ; Class 6
Fare=2, 2, 35           ; Class 2, AVI (type 2)
Fare=3, 2, 45           ; Class 3
Fare=4, 2, 55           ; Class 4
Fare=5, 2, 65           ; Class 5
Fare=6, 2, 75           ; Class 6
Fare=2, 8, 25           ; Class 2, Credit AVI (type 8)
Fare=3, 8, 25           ; Class 3
Fare=4, 8, 25           ; Class 4
Fare=5, 8, 25           ; Class 5
Fare=6, 8, 25           ; Class 6

[FareStructure]
FareID=2                 ; $0.50 starting fare
Fare=2, 1, 50           ; Class 2, cash (type 1)
Fare=3, 1, 75           ; Class 3
Fare=4, 1, 100          ; Class 4
Fare=5, 1, 125          ; Class 5
Fare=6, 1, 150          ; Class 6
Fare=2, 2, 35           ; Class 2, AVI (type 2)
Fare=3, 2, 45           ; Class 3
Fare=4, 2, 55           ; Class 4
Fare=5, 2, 65           ; Class 5
Fare=6, 2, 75           ; Class 6
                        ; Credit AVI (type 8) set to zero

[FareStructure]
FareID=3                 ; $0.65 strarting fare
Fare=2, 1, 65           ; Class 2, cash (type 1)
Fare=3, 1, 95           ; Class 3
Fare=4, 1, 125          ; Class 4
Fare=5, 1, 150          ; Class 5
Fare=6, 1, 200          ; Class 6
Fare=2, 2, 45           ; Class 2, AVI (type 2)
Fare=3, 2, 55           ; Class 3
Fare=4, 2, 65           ; Class 4
Fare=5, 2, 75           ; Class 5
Fare=6, 2, 85           ; Class 6
                        ; Credit AVI (type 8) set to zero

[Schedule-Fare]
ScheduleID=1             ; Fixed rate for all 7 days
Period=SUN, 00:00, SUN, 00:00, 1

[Schedule-Fare]
ScheduleID=2             ; Variable rate AM rush
Period=SUN, 00:00, MON, 05:00, 2
Period=MON, 05:00, MON, 09:00, 3 ; 5am-9am
Period=MON, 09:00, TUE, 05:00, 2
```

```
Period=TUE, 05:00, TUE, 09:00, 3 ; 5am-9am
Period=TUE, 09:00, WED, 05:00, 2
Period=WED, 05:00, WED, 09:00, 3 ; 5am-9am
Period=WED, 09:00, THU, 05:00, 2
Period=THU, 05:00, THU, 09:00, 3 ; 5am-9am
Period=THU, 09:00, FRI, 05:00, 2
Period=FRI, 05:00, FRI, 09:00, 3 ; 5am-9am
Period=FRI, 09:00, SUN, 00:00, 2
```

```
[Schedule-Fare]
ScheduleID=3 ; Variable rate PM rush
Period=SUN, 00:00, MON, 16:00, 2
Period=MON, 16:00, MON, 19:00, 3 ; 4pm-7pm
Period=MON, 19:00, TUE, 16:00, 2
Period=TUE, 16:00, TUE, 19:00, 3 ; 4pm-7pm
Period=TUE, 19:00, WED, 16:00, 2
Period=WED, 16:00, WED, 19:00, 3 ; 4pm-7pm
Period=WED, 19:00, THU, 16:00, 2
Period=THU, 16:00, THU, 19:00, 3 ; 4pm-7pm
Period=THU, 19:00, FRI, 16:00, 2
Period=FRI, 16:00, FRI, 19:00, 3 ; 4pm-7pm
Period=FRI, 19:00, SUN, 00:00, 2
```

```
[Plaza-Schedule]
Plaza=ABC, MLP1, N, 01/01/2004 00:00, 99/99/9999 23:59, 1
Plaza=ABC, MLP1, N, 07/04/2006 00:00, 07/05/2006 00:00, 2
Plaza=ABC, MLP1, S, 01/01/2004 00:00, 99/99/9999 23:59, 3
```

Likewise, an equivalent sample Lane Variable Rate Schedule File may be produced as follows.

```
; Created 10/01/2004 08:00

[Plaza-Schedule]
Plaza=ABC, MLP1, N, 01/01/2004 00:00, 99/99/9999 23:59, 1
Plaza=ABC, MLP1, N, 07/04/2006 00:00, 07/05/2006 00:00, 2
Plaza=ABC, MLP1, S, 01/01/2004 00:00, 99/99/9999 23:59, 3

[Schedule-Fare]
ScheduleID=1 ; Fixed rate for all 7 days
Period=SUN, 00:00, SUN, 00:00, 1

[Schedule-Fare]
ScheduleID=2 ; Variable rate AM rush
Period=SUN, 00:00, MON, 05:00, 2
Period=MON, 05:00, MON, 09:00, 3 ; 5am-9am
Period=MON, 09:00, TUE, 05:00, 2
Period=TUE, 05:00, TUE, 09:00, 3 ; 5am-9am
Period=TUE, 09:00, WED, 05:00, 2
Period=WED, 05:00, WED, 09:00, 3 ; 5am-9am
Period=WED, 09:00, THU, 05:00, 2
Period=THU, 05:00, THU, 09:00, 3 ; 5am-9am
Period=THU, 09:00, FRI, 05:00, 2
Period=FRI, 05:00, FRI, 09:00, 3 ; 5am-9am
Period=FRI, 09:00, SUN, 00:00, 2

[Schedule-Fare]
ScheduleID=3 ; Variable rate PM rush
Period=SUN, 00:00, MON, 16:00, 2
Period=MON, 16:00, MON, 19:00, 3 ; 4pm-7pm
Period=MON, 19:00, TUE, 16:00, 2
Period=TUE, 16:00, TUE, 19:00, 3 ; 4pm-7pm
Period=TUE, 19:00, WED, 16:00, 2
Period=WED, 16:00, WED, 19:00, 3 ; 4pm-7pm
Period=WED, 19:00, THU, 16:00, 2
Period=THU, 16:00, THU, 19:00, 3 ; 4pm-7pm
Period=THU, 19:00, FRI, 16:00, 2
Period=FRI, 16:00, FRI, 19:00, 3 ; 4pm-7pm
Period=FRI, 19:00, SUN, 00:00, 2

[FareStructure]
FareID=1
Fare=2, 1, 50 ; Class 2, cash (type 1)
Fare=3, 1, 75 ; Class 3
Fare=4, 1, 100 ; Class 4
Fare=5, 1, 125 ; Class 5
Fare=6, 1, 150 ; Class 6
Fare=2, 2, 35 ; Class 2, AVI (type 2)
Fare=3, 2, 45 ; Class 3
Fare=4, 2, 55 ; Class 4
Fare=5, 2, 65 ; Class 5
Fare=6, 2, 75 ; Class 6
Fare=2, 8, 25 ; Class 2, Credit AVI (type 8)
Fare=3, 8, 25 ; Class 3
Fare=4, 8, 25 ; Class 4
Fare=5, 8, 25 ; Class 5
Fare=6, 8, 25 ; Class 6

[FareStructure]
FareID=2 ; $0.50 starting fare
```

```
Fare=2, 1, 50 ; Class 2, cash (type 1)
Fare=3, 1, 75 ; Class 3
Fare=4, 1, 100 ; Class 4
Fare=5, 1, 125 ; Class 5
Fare=6, 1, 150 ; Class 6
Fare=2, 2, 35 ; Class 2, AVI (type 2)
Fare=3, 2, 45 ; Class 3
Fare=4, 2, 55 ; Class 4
Fare=5, 2, 65 ; Class 5
Fare=6, 2, 75 ; Class 6
; Credit AVI (type 8) set to zero
```

```
[FareStructure]
FareID=3 ; $0.65 strarting fare
Fare=2, 1, 65 ; Class 2, cash (type 1)
Fare=3, 1, 95 ; Class 3
Fare=4, 1, 125 ; Class 4
Fare=5, 1, 150 ; Class 5
Fare=6, 1, 200 ; Class 6
Fare=2, 2, 45 ; Class 2, AVI (type 2)
Fare=3, 2, 55 ; Class 3
Fare=4, 2, 65 ; Class 4
Fare=5, 2, 75 ; Class 5
Fare=6, 2, 85 ; Class 6
; Credit AVI (type 8) set to zero
```

Appendix F Domains

F.1 AckCodes

Ack Code	Ack Message
0	File Received Successfully
1	No Rate Schedule file is received.
2	No TVL Full File is received
3	No TVL Incremental File is received
11	File cannot be unzipped
12	File cannot be processed
13	File format is not correct
101	The record is blank
102	The TransponderValidationList is missing
103	The FileGenerateTime is missing
104	The TotalRecordCount is missing
105	The CSCClassification is missing
106	The TransponderID is missing
107	The TransponderAgencyCode is missing
108	The TransponderStatusCode is missing
109	The total record count does not match the Header
201	The record is blank
211	The [FareStructure] section is wrong
212	The [FareStructure] – FareID is wrong
213	The [FareStructure] – Fare is wrong

214	The [FareStructure] – class is wrong
215	The [FareStructure] – payment-type is wrong
216	The [FareStructure] – fare-amount is wrong
217	The [FareStructure] – per-axle-fare is wrong
221	The [Schedule-Fare] section is wrong
222	The [Schedule-Fare] – ScheduleID is wrong
223	The [Schedule-Fare] – Period is wrong
224	The [Schedule-Fare] – start-day is wrong
225	The [Schedule-Fare] – start-time is wrong
226	The [Schedule-Fare] – end-day is wrong
227	The [Schedule-Fare] – end-time is wrong
228	The [Schedule-Fare] – fare-id is wrong
231	The [Plaza-Schedule] section is wrong
232	The [Plaza-Schedule] – Plaza is wrong
233	The [Plaza-Schedule] – facility code is wrong
234	The [Plaza-Schedule] – plaza code is wrong
235	The [Plaza-Schedule] – direction is wrong
236	The [Plaza-Schedule] - effective date/time is wrong
237	The [Plaza-Schedule] – expiry date/time is wrong
238	The [Plaza-Schedule] – schedule-id is wrong
Ack Code Range	Ack Message
0	Success

1 - 100	File transfer error
101 – 200	TVL specific errors
201 - 300	Rate Schedule specific errors
999	Unknown Error – Contact Maintenance Group

F.2 TransponderAgencyCodes

AgencyCode	Description
78	WSDOT
218	TNB
219	CVISN

F.3 DeclarationStates

DeclarationState	Description
0	Undefined
1	SOV
2	HOV 2+
3	HOV 3+
4	HOV 4+
5	Exempt

F.4 FacilityCodes

FacilityCode	Description
167N	SR167
167S	SR167
SR16	TNB
520E	SR520 East
520W	SR520 West

F.5 InterfaceNames

InterfaceName	Description
RATE	Rate Schedule

TVL	Transponder Validation List
TRXN	Transactions

F.6 LaneModes

LaneMode	Description
0	Lane closed
1	Error
4	AVI
5	Emergency
6	Lane Open
7	AVI and Photo Toll
8	Photo Toll Only
9	Cash

F.7 PlazaCodes

FacilityCode	PlazaCode	Description
167N	15SW	SR167 - 15th St SW HOT Lane
167N	SR18	SR167 - SR18 HOT Lane
167N	15NW	SR167 - 15th St NW HOT Lane
167N	S277	SR167 - S 277th St HOT Lane
167N	SR516	SR167 - SR 516 N HOT Lane
167N	S212	SR167 - S 212th St HOT Lane
167S	I405	SR167 - I405 HOT Lane
167S	SW41	SR167 - SW 41st St HOT Lane
167S	84AVE	SR167 - 84th Ave S HOT Lane
167S	SR516	SR167 - SR 516 S HOT Lane
SR16	TNB	SR16 Tacoma Narrows Bridge - Lane 01
SR16	TNB	SR16 Tacoma Narrows Bridge - Lane 02
SR16	TNB	SR16 Tacoma Narrows Bridge - Lane 03
SR16	TNB	SR16 Tacoma Narrows Bridge - Lane 04
SR16	TNB	SR16 Tacoma Narrows Bridge - Lane 05
SR16	TNB	SR16 Tacoma Narrows Bridge - Lane 06
SR16	TNB	SR16 Tacoma Narrows Bridge - Lane 07
SR16	TNB	SR16 Tacoma Narrows Bridge - Lane 08
SR16	TNB	SR16 Tacoma Narrows Bridge - Lane 09
SR16	TNB	SR16 TNB 24th Street On-Ramp - Lane 10

520E	520EB	SR520 Bridge Eastbound - Lane 01
520E	520EB	SR520 Bridge Eastbound - Lane 02
520W	520WB	SR520 Bridge Westbound - Lane 01
520W	520WB	SR520 Bridge Westbound - Lane 02

F.8 TCSStatusCodes

TCSStatusCode	Description
1	All equipment functioning normal
2	Some equipment in degraded state

F.9 TransactionTypes

TransactionType	Description
2	Transponder (AVI)
1	Photo Enforced
3	Cash
4	Photo Toll
99	Not Available

F.10 TransponderProtocols

TransponderProtocol	Description
SeGo	Super eGo
ASTM	ASTM
ATA	ATA
6C	ISO 18000 6C

F.11 TransponderStatusCodes

TransponderStatusCode	Description
0	No Status
1	Valid
2	Low Balance
3	Invalid
4	Stolen
5	Lost
6	No Balance

F.12 UnusualOccurrenceCodes

UnusualOccurrenceCode	Description
1	Authorized Run Through
2	Run Through
3	Underclass
4	Class Mismatch
6	Invalid Transponder
7	Lost/Stolen Transponder
9	Insufficient Funds
12	Overclass
15	Lane Reset
17	Lane Update
18	Purged VSR
23	No Balance Transponder

F.13 VehicleClassTypes

VehicleClassType	Description
0	No Vehicle Class
1	Class 1
2	Class 2
3	Class 3
4	Class 4
5	Class 5
6	Class 6
7	Class 7
8	Class 8
9	Class 9
10	Class 10

Appendix G Complex Types

G.1 TransponderType

The TransponderType is an XML complex type describing the detailed features of a transponder.

The XML tags are described in the following table:

TagName	Required?	Data Type	Notes
TransponderID	Required	string	Transponder Serial Number <i>Format:</i> 10 digits with leading zeros <i>Range:</i> 0000000000 to 2147483647
IssuingAgencyCode	Required	string	Agency code of the issuing agency <i>Domain:</i> AgencyCodes
TransponderProtocol	Required	string	Transponder protocol <i>Domain:</i> TransponderProtocols
TransponderStatusCode	Required	string	Transponder status at the time the transaction was read in the lane. This status is pushed to the lanes in the transponder validation list file(s). <i>Domain:</i> TransponderStatusCodes
DeclarationState	Optional	string	Patrons may declare their state in HOT lanes. <i>Domain:</i> DeclarationStates
LowBattery	Optional	boolean	Identifies the condition of the battery in the transponder.

G.2 ImageListType

The ImageListType is an XML complex type describing the detailed features of a list of images.

The XML tags are described in the following table:

TagName	Required?	Data Type	Notes
ImageLocation	Required	string	Directory containing the images.
SelectedImageName	Optional	string	ImageName that was chosen by the TCS vendor as the preferred image. In most cases, this will be the image with the best OCR results, but in some cases, the TCS will follow different rules. For instance, the SR520 SOW states that they must choose the front shot image if the OCR results do not match between

			the front and the rear, and if the AxleCount is 3 or greater.
SelectedROIImageName	Optional	string	ImageName of the ROI that corresponds with the chosen Image.
Image minOccurs="1" maxOccurs="unbounded"			
ImageName	Required	string	File name of the image.
ImageType	Required	string	<i>Domain:</i> ROI – Region of Interest FSI – Front Shot Image BSI – Back Shot Image EFI – Entry Front Shot Image (Hot Lanes Only) EBI – Entry Back Shot Image (Hot Lanes Only)
OCRPlateNumber	Optional	string	License plate number as identified by the OCR software
OCRPlateState	Optional	string	License plate state as identified by the OCR software
OCRPlateType	Optional	string	License plate type as identified by the OCR software
OCRPlateNumberConfidence	Optional	int	Confidence percentage assigned by the OCR software for license plate number. <i>Range: 0 to 99</i>
OCRPlateStateConfidence	Optional	int	Confidence percentage assigned by the OCR software for license plate state. <i>Range: 0 to 99</i>
OCRPlateTypeConfidence	Optional	int	Confidence percentage assigned by the OCR software for license plate type. <i>Range: 0 to 99</i>

Appendix H Response Codes

Response Code	Response Message	Troubleshooting Tips
000	Transaction received successfully	No troubleshooting. Rejoice instead.
101	System is down for maintenance	Check for scheduled maintenance. If necessary, call the CSC help desk to verify. Try to determine when the system will become available. In the meantime, queue transactions so they can be submitted when the system comes back up.
102	General error (undefined)	The web service failed for an unexpected reason. Additional information will be provided in the ResponseMessage data element. Use the ResponseMessage information to troubleshoot. Report the error to the CSC help desk.
103	Invalid data in field "TagName"	Verify that the data was submitted correctly. Confirm by comparing to this ICD. If the data was specified correctly, report to the CSC help desk.
104	Unknown plaza "PlazaCode"	The unknown PlazaCode will be specified in the ResponseMessage data element. Verify that the correct PlazaCode was submitted.
105	Unknown Lane "LaneNumber"	The unknown LaneNumber will be specified in the ResponseMessage data element. Verify that the correct LaneNumber was submitted.
106	Missing required field, "TagName"	Correct the code to include the missing data element.
107	Transaction older than 60 days	Verify that the TransactionDateTime is correct. If it is correct, try to determine why old data is being submitted.
108	Duplicate transaction previously received with ResponseCode "xxx".	This response is expected if the TCS vendor resubmits old transactions, where some were successfully transmitted, but some were not. If this happens on an ongoing basis, the TCS vendor needs to track down the information that is being submitted more than once.
109	XML format error.	If possible, the response message details will be specified in the ResponseMessage field.

APPENDIX 13.2 – SR 167 TCS – TO – SIMMS ICD

This page intentionally left blank

This page intentionally left blank.

Washington State Department of Transportation

Tacoma Narrows / SR167 Hot Lane Integration

Interface Control Document



9440 Carroll Park Drive, Suite 150
San Diego, CA 92121-5201

August 13, 2007

840D0036

This page intentionally left blank.

Table of Contents

1. INTRODUCTION	1
1.1 OVERVIEW	1
1.2 PURPOSE	1
1.3 STRUCTURE.....	1
2. GENERAL INFORMATION	3
2.1 DATA EXCHANGES	3
2.2 TRANSMISSION REQUIREMENTS	3
2.3 TRANSACTION PROCESSING	4
2.4 REPORTING REQUIREMENTS	4
2.5 FILE REQUIREMENTS	4
3. DATA EXCHANGES	6
3.1 TAG STATUS FILE	6
3.2 TRANSACTION FILE	9
3.3 TRANSFER COMPLETE FILE	12
3.4 ACKNOWLEDGEMENT FILE	14
4. FILE TRANSFER FLOW	16
4.1 TAG STATUS FILES	16
4.2 TRANSACTION FILES	16
5. FAILURE PATH	17
6. SECURITY	18
6.1 INTRODUCTION	18
6.2 IN TRANSIT	18
6.3 IN STORAGE	18
7. DATA ELEMENTS	20
APPENDIX A TAG STATUS FILE XML SCHEMA.....	23
APPENDIX B TRANSACTION FILE XML SCHEMA.....	25
APPENDIX C TRANSFER COMPLETE FILE XML SCHEMA.....	26
APPENDIX D ACKNOWLEDGEMENT FILE XML SCHEMA	27

This page intentionally left blank.

1. INTRODUCTION

1.1 Overview

The Washington State Department of Transportation (WSDOT) will be implementing the following two toll collection systems in the near future:

- Tacoma Narrows Bridge
- SR167 high occupancy toll (HOT) lanes

Both systems will be able to collect tolls electronically using automatic vehicle identification (AVI) transponders. The sole method for collecting tolls on SR167 system will be through the use of AVI transponders on single occupancy vehicles. However, the SR167 system will not include a customer service center.

The Tacoma Narrows Bridge system includes a fully functional customer service and violations processing center. The WSDOT plans to enable AVI accounts maintained at this service center to be able to be used for paying tolls on SR167.

1.2 Purpose

To ensure a successful integration of the SR167 HOT lanes with the Tacoma Narrows Bridge TCAAS, the purpose of this Interface Control Document (ICD) is to:

- Specify a non proprietary solution for data exchanges between SR167 and the TCAAS
- Specify a solution that can be reused by WSDOT for other high occupancy toll facilities in Washington that may want to use the TCAAS for customer service functions
- Specify all data exchanges necessary to support appropriate customer service functions by staff at the Tacoma Narrows Bridge Customer Service Center
- Specify all data exchanges necessary to support high occupancy lane operation by SR167

1.3 Structure

Following this introduction, the document has been split into the following six sections:

- General information – describing general processes that apply to all data exchanges
- Data exchanges – describes each data exchange as well as the layout and detail description of data included in the file

- File Transfer Flow - describes the process/methods/communications protocols/timings for the transmission of data/files
- Failure Path – describes processes for dealing with failed file transfers or failed processing of files
- Security requirements – describes processes/methods required for secure transmission of data/files
- Data Elements - provides additional definition of each data element referenced in the data exchanges

2. GENERAL INFORMATION

2.1 Data Exchanges

The Data Exchanges defined in this Interface Control Document are

Data Exchange	File Extension	Usage
Tag Status Data	FTSF	Created each day by the TCAAS to include all valid tags.
HOT Lane Transaction Data	FTXF	Created each day by the Facility (SR167) to include all transactions that need to be posted to an account maintained by TCAAS.
Transfer Complete	CHK	Created by the entity transferring a file to indicate that the file transfer is complete and the corresponding file can be validated and processed. The absence of this file indicates that the file transfer has not been completed.
Acknowledgment	ACK	Created by the entity downloading the file from the FTP server. It is used to confirm that the file has been validated. Error codes within the file provide additional information to the TCAAS or Facility.

An FTP server will be managed by the Tacoma Narrows CSC to enable the TCAAS to receive data files from Facilities. This same FTP server will also include files from TCAAS that each Facility will be required to download.

2.2 Transmission Requirements

File transfers between the Facility and the TCAAS FTP server will be completed using the file transfer protocol (FTP).

The schedule for transmitting files between TCAAS and Facilities are

Data Exchange	Transmission Frequency	Notes
Tag Status Data	Twice per day	1) After nightly replenishment and prior to 04:00AM. 2) Between 12:00PM and 1:00PM
HOT Lane Transaction Data	Once per day	Between 00:00 and 01:00.
Transfer Complete	Varies	One for each file transfer
Acknowledgment	Varies	One for each file transfer

2.3 Transaction Processing

Transactions from a Facility shall be sent to TCAAS in a timely manner. The TCAAS shall reject original transactions that are older than 60 days.

Transactions that cannot be posted to accounts will result in the facility not receiving the amount due for the transaction.

2.4 Reporting Requirements

The TCAAS will provide the financial reporting for the SR167 HOT lanes. Several reports will be available in TCAAS to identify the count of toll transactions that the TCAAS successfully posted for that Facility for a particular revenue date. Tolls, fees or other deductions will not be included in this report.

An additional report (Transponder Transaction Exception Report) will also be available to the Facility via TCAAS to identify those transactions that were not posted to an account. This report represents the amount of toll transactions that the Facility will not receive for a particular revenue date.

2.5 File Requirements

2.5.1 Format

Files will be well-formed XML files. The following rules will be applied for using XML tags:

- XML tags must be nested properly
- XML tags are case sensitive. Therefore, <tag> and <Tag> are treated as different/distinct.
- XML tags must have a beginning and an end
- Each tag must be enclosed in < and >
- Optional tags do not need to be included. However if one of the detail records contains optional tags then all other detail records must include these tags even if there is no data for the tag.

2.5.2 Top Level (Root) Tags

The Top Level Tag in all files will include a description of the file type and the version number of the ICD to which the data complies. Spaces will be replaced by the underscore (“_”) character.

For example, the transaction file that complies with version 1.2 of this document will have a top level tag as follows: <TRANSACTION_FILE_1.2>

2.5.3 Dates

Any dates used in file names of data elements will utilize the 24 hour clock. For example, 3:05pm will always be written as 15:05.

2.5.4 File Naming Conventions

Only uppercase characters shall be used in file names.

Compression

All files transmitted to the FTP server (except for the Transfer Complete and Acknowledgement Files) shall be compressed using a standard Lempel-Zif compression algorithm to yield a compression rate of at least 75%. The extension of these files will all be converted from [File Name].[File Type] to [File Name]_[File Type].ZIP.

2.5.5 Encryption

All files transmitted to the FTP server (except for the Transfer Complete file) shall be encrypted after they have been compressed and before they are transmitted to the FTP server. The security requirements for files are described in a later section of this document.

2.5.6 Archiving

All files on the FTP server will remain available for download for a period of seven (days). Files with an upload date that is older than seven (7) days will be moved from the FTP server to an archive folder. This archive folder will only be accessible by TCAAS.

Files in the FTP archive folder shall remain accessible via online storage for at least 60 days. Files shall remain accessible via offline storage (such as a tape backup) for at least 360 days after being deleted from the archive folder. These files will only be available to use in case of disputes between TCAAS and a Facility.

3. DATA EXCHANGES

3.1 Tag Status File

3.1.1 Description

The Tag Status File is created by TCAAS and transmitted to a Facility for processing. It will contain the tag status data for all tags associated with an account maintained at the TCAAS. Additional Tag information will also be included to enable the Facility to identify if the tag is a revenue or non-revenue tag.

3.1.2 Naming Convention

[FILE_DATE_TIME].FTSF

Example: For a Tag Status file created by at 02:00:05 on June 19, 2006 the name of the file be 20060619020005.FTSF.

3.1.3 Layout

```
<TAG_STATUS_FILE_1.0>
<HEADER>
  <FILE_DATE_TIME></FILE_DATE_TIME>
  <TOTAL_TAG_COUNT></TOTAL_TAG_COUNT>
  <TOTAL_TAG_COUNT_STATUS_0></TOTAL_TAG_COUNT_STATUS_0>
  <TOTAL_TAG_COUNT_STATUS_1></TOTAL_TAG_COUNT_STATUS_1>
  <TOTAL_TAG_COUNT_STATUS_2></TOTAL_TAG_COUNT_STATUS_2>
  <TOTAL_TAG_COUNT_STATUS_3></TOTAL_TAG_COUNT_STATUS_3>
  <TOTAL_TAG_COUNT_STATUS_4></TOTAL_TAG_COUNT_STATUS_4>
  <TOTAL_TAG_COUNT_STATUS_5></TOTAL_TAG_COUNT_STATUS_5>
</HEADER>
<DETAIL_DATA>
  <TAG>
    <RECORD_ID></RECORD_ID>
    <TAG_AGENCY_ID></TAG_AGENCY_ID>
    <TAG_PROTOCOL></TAG_PROTOCOL>
    <TAG_REVENUE_TYPE></TAG_REVENUE_TYPE>
    <TAG_NUMBER></TAG_NUMBER>
    <TAG_HW_SERIAL_NUMBER></TAG_HW_SERIAL_NUMBER>
    <TAG_CLASS></TAG_CLASS>
    <TAG_STATUS></TAG_STATUS>
  </TAG>
</DETAIL_DATA>
<FOOTER>
</FOOTER>
</TAG_STATUS_FILE_1.0>
```

3.1.4 Data Elements

3.1.4.1 Top Level (Root) Tag

The file description used in the top level xml tag will be <TAG_STATUS_FILE_x.x> where x.x is the version number of the ICD with which the file will comply.

3.1.4.2 Header

Each file will contain a header record containing data applicable to all detailed records and providing summary data to be used to verify file integrity. Listed below are the data elements for the <HEADER> record in a Tag Status File.

Data Element Name	Mandatory
FILE_DATE_TIME	Yes
TAG_TOTAL_COUNT	Yes
TAG_COUNT_STATUS_0	Yes
TAG_COUNT_STATUS_1	Yes
TAG_COUNT_STATUS_2	Yes
TAG_COUNT_STATUS_3	Yes
TAG_COUNT_STATUS_4	Yes
TAG_COUNT_STATUS_5	Yes

3.1.4.3 Detail Data

Each Tag record will be contained within a <TAG> record. Listed below are the data elements for the <TAG> record.

Data Element Name	Mandatory
RECORD_ID	Yes
TAG_AGENCY_ID	Yes
TAG_PROTOCOL	Yes
TAG_REVENUE_TYPE	Yes
TAG_NUMBER	Yes
TAG_HW_SERIAL_NUMBER	No
TAG_CLASS	Yes
TAG_STATUS	Yes

3.1.4.4 Footer

Each file will contain a footer record with no required data elements. Future enhancements to this interface control document may necessitate the inclusion of tags in this record. However, for this version of the interface control document, no tags are required to be included in the footer record.

3.1.5 Example

```
<?xml version="1.0"?>
<TAG_STATUS_FILE_1.0 xmlns="http://localhost/SR167"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation = "http://localhost/SR167
http://localhost/SR167/tagStatusFileSchema.xsd">
<HEADER>
    <FILE_DATE_TIME>2006-07-27T00:10:10</FILE_DATE_TIME>
    <TOTAL_TAG_COUNT>2</TOTAL_TAG_COUNT>
    <TOTAL_TAG_COUNT_STATUS_0>0</TOTAL_TAG_COUNT_STATUS_0>
    <TOTAL_TAG_COUNT_STATUS_1>1</TOTAL_TAG_COUNT_STATUS_1>
    <TOTAL_TAG_COUNT_STATUS_2>1</TOTAL_TAG_COUNT_STATUS_2>
    <TOTAL_TAG_COUNT_STATUS_3>0</TOTAL_TAG_COUNT_STATUS_3>
    <TOTAL_TAG_COUNT_STATUS_4>0</TOTAL_TAG_COUNT_STATUS_4>
    <TOTAL_TAG_COUNT_STATUS_5>0</TOTAL_TAG_COUNT_STATUS_5>
</HEADER>
<DETAIL_DATA>
    <TAG>
        <RECORD_ID>1</RECORD_ID>
        <TAG_AGENCY_ID>218</TAG_AGENCY_ID>
        <TAG_PROTOCOL>SEGO</TAG_PROTOCOL>
        <TAG_REVENUE_TYPE>REVENUE</TAG_REVENUE_TYPE>
        <TAG_NUMBER>1012</TAG_NUMBER>
        <TAG_HW_SERIAL_NUMBER>010000DA00700112</TAG_HW_SERIAL_NUMBER>
        <TAG_CLASS>2</TAG_CLASS>
        <TAG_STATUS>1</TAG_STATUS>
    </TAG>
    <TAG>
        <RECORD_ID>2</RECORD_ID>
        <TAG_AGENCY_ID>219</TAG_AGENCY_ID>
        <TAG_PROTOCOL>CVISN</TAG_PROTOCOL>
        <TAG_REVENUE_TYPE>NONREVENUE</TAG_REVENUE_TYPE>
        <TAG_NUMBER>671861690</TAG_NUMBER>
        <TAG_HW_SERIAL_NUMBER></TAG_HW_SERIAL_NUMBER>
        <TAG_CLASS>2</TAG_CLASS>
        <TAG_STATUS>2</TAG_STATUS>
    </TAG>
</DETAIL_DATA>
<FOOTER>
</FOOTER>
</TAG_STATUS_FILE_1.0>
```

3.2 Transaction File

3.2.1 Description

The Transaction File is created by the Facility and transmitted to the TCAAS for processing. It will contain all the required transaction data occurring at the Facility for tags belonging to the TCAAS so that TCAAS can apply the transactions against the appropriate customer account.

3.2.2 Naming Convention

[FAC_ID]_[FILE_DATE_TIME].FTXF

Example: For a Transaction File created by Facility 45 at 00:43:21 on November 31, 2006 the name of the file be 045_20061131004321.FTXF.

3.2.3 Layout

```
<TRANSACTION_FILE_1.0>
  <HEADER>
    <FAC_ID></FAC_ID>
    <FILE_ID></FILE_ID>
    <FILE_DATE_TIME></FILE_DATE_TIME>
    <TRANSACTION_COUNT></TRANSACTION_COUNT>
    <TRANSACTION_SUM></TRANSACTION_SUM>
  </HEADER>
  <DETAIL_DATA>
    <TRANSACTION>
      <RECORD_ID></RECORD_ID>
      <TRANSACTION_ID></TRANSACTION_ID>
      <TAG_AGENCY_ID></TAG_AGENCY_ID>
      <TAG_NUMBER></TAG_NUMBER>
      <ENTRY_TRANS_DATE_TIME></ENTRY_TRANS_DATE_TIME>
      <ENTRY_PLAZA_NO></ENTRY_PLAZA_NO>
      <ENTRY_LANE_NO></ENTRY_LANE_NO>
      <TX_CLASS></TX_CLASS>
      <AVC_CLASS></AVC_CLASS>
      <TAG_CLASS></TAG_CLASS>
      <TOLL></TOLL>
      <SPECIAL_RULE_APPLIED></SPECIAL_RULE_APPLIED>
      <TRIP_SPEED></TRIP_SPEED>
    </TRANSACTION>
  </DETAIL_DATA>
</FOOTER>
</FOOTER>
</TRANSACTION_FILE_1.0>
```

3.2.4 Data Elements

3.2.4.1 Top Level (Root) Tag

The file description used in the top level xml tag will be <TRANSACTION_FILE_x.x> where x.x is the version number of the ICD with which the file will comply.

3.2.4.2 Header

Each file will contain a header record containing data applicable to all detailed records and providing summary data to be used to verify file integrity. Listed below are the data elements for the <HEADER> record in a Transaction File.

Data Element Name	Mandatory
FAC_ID	Yes
FILE_ID	Yes
FILE_DATE_TIME	Yes
TRANSACTION_COUNT	Yes
TRANSACTION_SUM	Yes

3.2.4.3 Detail Data

Each transaction record will be contained within a <TRANSACTION> record. Listed below are the data elements for the <TRANSACTION> record.

Data Element Name	Mandatory
RECORD_ID	Yes
TRANSACTION_ID	Yes
TAG_AGENCY_ID	Yes
TAG_NUMBER	Yes
ENTRY_TRANS_DATE_TIME	Yes
ENTRY_PLAZA_NO	Yes
ENTRY_LANE_NO	Yes
TX_CLASS	Yes
AVC_CLASS	No
TAG_CLASS	No
TOLL	Yes
SPECIAL_RULE_APPLIED	Yes
TRIP_SPEED	No

3.2.4.4 Footer

Each file will contain a footer record with no required data elements. Future enhancements to this interface control document may necessitate the inclusion of tags in this record. However, for this version of the interface control document, no tags are required to be included in the footer record.

3.2.5 Example

```
<?xml version="1.0"?>
<TRANSACTION_FILE_1.0 xmlns="http://localhost/SR167"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation =
```

```

"http://localhost/SR167 http://localhost/SR167/transactionFileSchema.xsd">
  <HEADER>
    <FAC_ID>100</FAC_ID>
    <FILE_ID>987654321</FILE_ID>
    <FILE_DATE_TIME>2006-06-19T00:43:21</FILE_DATE_TIME>
    <TRANSACTION_COUNT>2</TRANSACTION_COUNT>
    <TRANSACTION_SUM>1.50</TRANSACTION_SUM>
  </HEADER>
  <DETAIL_DATA>
    <TRANSACTION>
      <RECORD_ID>1</RECORD_ID>
      <TRANSACTION_ID>123456788</TRANSACTION_ID>
      <TAG_AGENCY_ID>218</TAG_AGENCY_ID>
      <TAG_NUMBER>1012</TAG_NUMBER>
      <ENTRY_TRANS_DATE_TIME>2006-06-18T09:12:15</ENTRY_TRANS_DATE_TIME>
      <ENTRY_PLAZA_NO>102</ENTRY_PLAZA_NO>
      <ENTRY_LANE_NO>1</ENTRY_LANE_NO>
      <TX_CLASS>1</TX_CLASS>
      <AVC_CLASS>1</AVC_CLASS>
      <TAG_CLASS>1</TAG_CLASS>
      <TOLL>1.50</TOLL>
      <SPECIAL_RULE_APPLIED>SR167-Y</SPECIAL_RULE_APPLIED>
      <TRIP_SPEED>75</TRIP_SPEED>
    </TRANSACTION>
    <TRANSACTION>
      <RECORD_ID>2</RECORD_ID>
      <TRANSACTION_ID>123456789</TRANSACTION_ID>
      <TAG_AGENCY_ID>219</TAG_AGENCY_ID>
      <TAG_NUMBER>671861690</TAG_NUMBER>
      <ENTRY_TRANS_DATE_TIME>2006-06-18T09:35:58</ENTRY_TRANS_DATE_TIME>
      <ENTRY_PLAZA_NO>102</ENTRY_PLAZA_NO>
      <ENTRY_LANE_NO>1</ENTRY_LANE_NO>
      <TX_CLASS>1</TX_CLASS>
      <AVC_CLASS>1</AVC_CLASS>
      <TAG_CLASS>1</TAG_CLASS>
      <TOLL>0.00</TOLL>
      <SPECIAL_RULE_APPLIED>SR167-Y</SPECIAL_RULE_APPLIED>
      <TRIP_SPEED>32</TRIP_SPEED>
    </TRANSACTION>
  </DETAIL_DATA>
</FOOTER>
</FOOTER>
</TRANSACTION_FILE_1.0>

```

3.3 Transfer Complete File

3.3.1 Description

The Transfer Complete File is created by the originating organization immediately after it has completed the upload of a file to the FTP server. It will contain the name of the file and type of file for which the file transfer has just been completed. The existence of this file on the FTP server will indicate that the file recently transferred is ready to be validated by the receiving organization.

3.3.2 Naming Convention

[FILENAME_FILETYPE].CHK

Example 1: For a Tag Status file (20060619020005.ftsf) that has been recently transferred in its entirety to a Facility, the name of the transfer complete file will be 20060619020005_FTSF.CHK.

Example 2: For a Transaction File (045_20061131124321.ftxf) that has been recently transferred in its entirety to the TCAAS, the name of the transfer complete file will be 045_20061131124321_FTXF.CHK.

3.3.3 Layout

```
<TRANSFER_COMPLETE_1.0>
  <DETAIL_DATA>
    <FILE>
      <FILENAME></FILENAME>
      <FILETYPE></FILETYPE>
    </FILE>
  </DETAIL_DATA>
</TRANSFER_COMPLETE_1.0>
```

3.3.4 Data Elements

3.3.4.1 Top Level (Root) Tag

The file description used in the top level xml tag will be <TRANSFER_COMPLETE_x.x> where x.x is the version number of the ICD with which the file will comply.

3.3.4.2 Detail Data

The one file record in the transfer complete file will be contained within a <FILE> record. Listed below are the data elements for the <FILE> record.

Data Element Name	Mandatory
FILENAME	Yes
FILETYPE	Yes

3.3.5 Example

```
<?xml version="1.0"?>
<TRANSFER_COMPLETE_1.0 xmlns="http://localhost/SR167"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation =
"http://localhost/SR167 http://localhost/SR167/chkFileSchema.xsd">
  <DETAIL_DATA>
    <FILE>
      <FILENAME>045_20061131124321</FILENAME>
      <FILETYPE>FTXF</FILETYPE>
    </FILE>
  </DETAIL_DATA>
</TRANSFER_COMPLETE_1.0>
```

3.4 Acknowledgement File

3.4.1 Description

The Acknowledgement File is created by the receiving organization after it has validated another file such as the Tag Status File or Transaction File. It will contain the name of the file and type of file that has been validated. Additional return code data will also be included to indicate success or failures involved in validating the file and total number of records included in the file.

3.4.2 Naming Convention

[FROM_FACILITY_ID]_[FILENAME]_[FILETYPE]_[FILE_DATE_TIME].ACK

Example 1: For a Tag Status file (20060619020005.ftsf) that has been validated by a Facility (045) and the acknowledgement file was created at 02:34:01 on June 19, 2006, the name of the acknowledgement file will be 045_20060619020005_FTSF_20060619023401.ACK.

Example 2: For a Transaction File (045_20061131004321.ftxf) that has been validated by TCAAS and the acknowledgement file was created at 00:52:35 on November 31, 2006, the name of the acknowledgement file will be 216_045_20061131004321_FTXF_20061131005235.ACK.

3.4.3 Layout

```
<ACKNOWLEDGEMENT_1.0>
<HEADER>
  <FILENAME></FILENAME>
  <FILETYPE></FILETYPE>
  <RETURN_CODE></RETURN_CODE>
  <NUM_RECORDS></NUM_RECORDS>
  <NUM_ISSUES></NUM_ISSUES>
</HEADER>
<DETAIL_DATA>
  <ERROR>
    <RECORD_ID></RECORD_ID>
    <DESCRIPTION></DESCRIPTION>
  </ERROR>
</DETAIL_DATA>
<FOOTER>
</FOOTER>
</ACKNOWLEDGEMENT_1.0>
```

3.4.4 Data Elements

3.4.4.1 Top Level (Root) Tag

The file description used in the top level xml tag will be <ACKNOWLEDGEMENT_x.x> where x.x is the version number of the ICD with which the file will comply.

3.4.4.2 Header

Each file will contain a header record containing summary data applicable to a particular . Listed below are the data elements for the <HEADER> record.

Data Element Name	Mandatory
FILENAME	Yes
FILETYPE	Yes
RETURN_CODE	Yes
NUM_RECORDS	Yes
NUM_ISSUES	Yes

3.4.4.3 Detail Data

Each error or informational warning will be contained within an <ISSUE> record. Listed below are the data elements for the <ISSUE> record.

Data Element Name	Mandatory
RECORD_ID	No
DESCRIPTION	No

3.4.4.4 Footer

Each file will contain a footer record with no required data elements. Future enhancements to this interface control document may necessitate the inclusion of tags in this record. However, for this version of the interface control document, no tags are required to be included in the footer record.

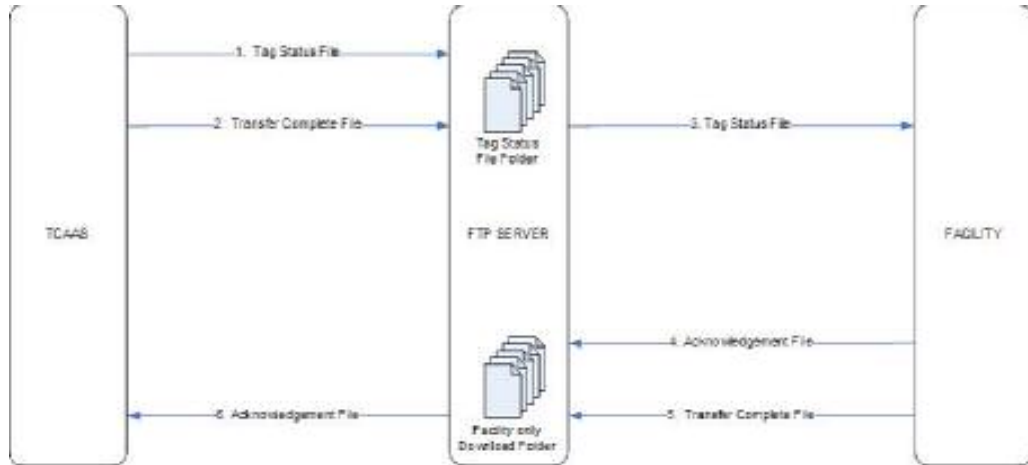
3.4.5 Example

```
<?xml version="1.0"?>
<ACKNOWLEDGEMENT_1.0 xmlns="http://localhost/SR167" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation = "http://localhost/SR167 http://localhost/SR167/ackFileSchema.xsd">
<HEADER>
  <FILENAME>20060619020005 </FILENAME>
  <FILETYPE>FTSF</FILETYPE>
  <RETURN_CODE>31</RETURN_CODE>
  <NUM_RECORDS>522</NUM_RECORDS>
  <NUM_ISSUES>1</NUM_ISSUES>
</HEADER>
<DETAIL_DATA>
  <ERROR>
    <RECORD_ID>1</RECORD_ID>
    <DESCRIPTION>TAG NUMBER FIELD FAILED VALIDATION</DESCRIPTION>
  </ERROR>
</DETAIL_DATA>
<FOOTER>
</FOOTER>
</ACKNOWLEDGEMENT_1.0>
```

4. FILE TRANSFER FLOW

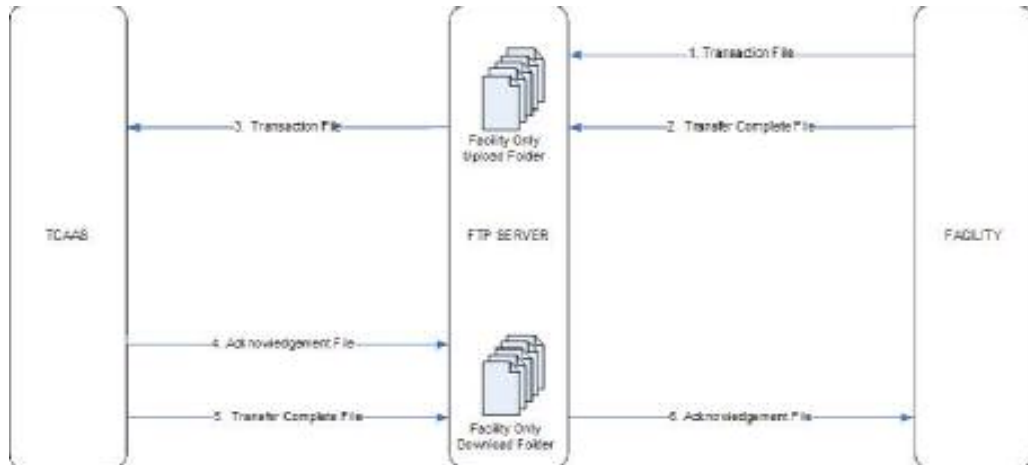
4.1 Tag Status Files

The process for transferring files Tag Status files is as follows:



4.2 Transaction Files

The process for transferring files Transaction files is as follows:



5. FAILURE PATH

The return code in the Acknowledgement file provides the organization that created the original file confirmation that the file has been successfully validated or an indication of the problem experienced by the receiving organization. The definition of the return codes and the actions to be taken when a return code is received are listed below.

Return Code	Description of problem	Actions to be taken
00	The file was successfully validated.	No further action necessary.
10	File cannot be decrypted.	<p>Verify originating organization can decrypt file. Resolve encryption problems at appropriate location.</p> <p>If the original file was encrypted badly then it can be deleted from FTP server and replaced by a correctly encrypted file with the same name.</p> <p>If the problem is related to the private key used for decryption, the original encrypted file will be revalidated and a new ACK file can be created.</p>
11	File cannot be uncompressed.	<p>Verify originating organization can decompress file. Resolve decompression at appropriate location.</p> <p>If the original file was compressed badly then it can be deleted from FTP server and replaced by a correctly compressed file with the same name.</p> <p>If the problem is related to the tool used for decompressing the file, the original compressed file will be revalidated and a new ACK file can be created.</p>
12	Badly formed XML file.	<p>The file will need to be recreated by the originating organization with correctly formed XML tags.</p> <p>The file can be resubmitted with the same name and File_ID value.</p>
13	Duplicate File. File with File_ID has already been validated.	Originating organization will need to create a new file with a non duplicate FILE_ID value.
20	Validation of the summary data in the header against the detailed record data was not successful. The file was not processed by TCAAS or the Facility.	<p>Originating organization will need to confirm the data in the original file is correct.</p> <p>Errors in the file can be corrected only if they relate to the summary data. In this case, the file can be resubmitted with the same name and File_ID value.</p> <p>If the errors are related to the detailed data, then the originating organization will be required to create a new file with different name and FILE_ID value.</p>
30	Transaction records contain invalid data. TCAAS will not post these to a customer account.	The Facility will be able to obtain the number of transactions not posted using the reports provided by TCAAS.
31	Tag records contain invalid data. Facility will not update status for tag record containing invalid data.	TCAAS will review tag status in original Tag Status File. If there is an error in the Tag Status File, a new file will be created and uploaded to the FTP server.
32	<p>Duplicate transaction.</p> <p>The file contains transactions with Transaction_ID values that have already been validated in another file.</p>	Originating organization will need to correct duplicate Transaction_ID values. The file cannot be resubmitted. A new file with non duplicate Transaction_ID values will need to be created and uploaded to the FTP server.

6. SECURITY

6.1 Introduction

The data exchanged between the TCAAS and facilities will be need to be secured from unauthorized access. This section of the ICD identifies the requirements for securing the data files from unauthorized access while in transit and in storage on an FTP server.

6.2 In Transit

To prevent unauthorized access to toll data via snooping or other methods designed to intercept data, all files except the Transfer Complete (CHK) files will be encrypted and decrypted using PGP encryption methodology based on shared and private keys. PGP version 9 will be the solution required to be used for key management and file encryption/decryption.

The following tables identifies which keys will be used for specific data

Data exchange	Encryption method	Decryption method
All files except Transaction Files	Shared key generated by TCAAS and distributed to all Facilities	Private key generated by TCAAS and distributed to all Facilities
Transaction Files	Unique shared key for specific facility generated by TCAAS	Unique private key for specific facility generated by TCAAS. Only distributed to specific facility.

These keys will be provided to the facilities on a CD 30 days prior to start of TCAAS-facility integration testing.

6.3 In Storage

6.3.1 FTP Server

The FTP server managed by TCAAS will include the following folder on the FTP server:

- Tag Status File Folder - contains files uploaded by TCAAS. TCAAS will have read/write/delete access, but will not have update access (to prevent file renaming). Access control will be restricted to read only permissions for all facilities.
- Facility Only Download Folder - contains files to be downloaded by only one Facility. There will be one of these folders for each Facility using TCAAS for CSC functions. Access control will be restricted to only one Facility with read only permissions. TCAAS will have read/write/delete access, but will not have update access (to prevent file renaming).
- Facility Only Upload Folder - contains files to be uploaded by only one Facility. There will be one of these folders for each Facility using TCAAS for CSC functions. Access control will be restricted to only one Facility with read/write permissions, but not update permissions (to prevent file renaming). TCAAS will have read/delete access permissions to this folder.

Internal, local user access to these folders will only be enabled for administrators and specific users with access to other secure toll data within the TCAAS network. Non Authorized users with access to the network on which the FTP server is located will not be given read permissions to any folders used by the FTP server or used to archive files exchanged between TCAAS and Facilities.

External Access to the login prompt for the FTP server will be restricted by IP address. If an IP address attempts to access the FTP server and it is not on the permitted list of IP addresses, access to the FTP server will not be allowed. A valid username/password combination will also be required to authenticate valid users and provide access to FTP folders.

7. DATA ELEMENTS

Data Element Name	Comments	XML Data Type	TCAAS DB Data Type																				
AVC_CLASS	Class of the vehicle according to the automatic vehicle classification system in the lane.	positiveInteger	SMALLINT, >0																				
DESCRIPTION	Description of error to assist Facility or TCAAS with debugging the problem.	string	VARCHAR(500)																				
ENTRY_LANE_NO	Lane number associated with transaction.	positiveInteger	SMALLINT, >0																				
ENTRY_PLAZA_NO	Plaza number associated with transaction.	positiveInteger	SMALLINT, >0																				
ENTRY_TRANS_DATE_TIME	YYYY-MM-DDThh:mm:ss.. Date/time transaction occurred in the lane.	dateTime	DATETIME																				
FAC_ID	Facility ID for remote locations using TCAAS for CSC functions.	positiveInteger	SMALLINT, >0																				
FILE_DATE_TIME	YYYY-MM-DDThh:mm:ss. Date file created.	dateTime	DATETIME																				
FILE_ID	Server generated transaction ID, which uniquely identifies the file.	positiveInteger	SMALLINT, >0																				
FILENAME	Name of file associated with Transfer Complete file.	string	VARCHAR(500)																				
FILETYPE	Type of file associated with Transfer Complete file.	string	VARCHAR(100)																				
NUM_RECORDS	Count of records (transactions or tag status) contained within the Acknowledgement file.	unsignedLong	SMALLINT, >0																				
NUM_ISSUES	Count of issues (errors or informational warnings) contained within the Acknowledgement file.	unsignedLong	SMALLINT, >0																				
RECORD_ID	Server generated record ID, which uniquely identifies a record within a specific file.	unsignedLong	SMALLINT, >0																				
RETURN_CODE	Issue (Error or informational warning) associated with the file. <table border="1"> <thead> <tr> <th>CODE</th> <th>DESCRIPTION</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>The file was successfully validated.</td> </tr> <tr> <td>10</td> <td>File cannot be decrypted.</td> </tr> <tr> <td>11</td> <td>File cannot be uncompressed.</td> </tr> <tr> <td>12</td> <td>Badly formed XML file.</td> </tr> <tr> <td>13</td> <td>Duplicate File_ID</td> </tr> <tr> <td>20</td> <td>Summary data validation failed.</td> </tr> <tr> <td>30</td> <td>Transaction records contain invalid data.</td> </tr> <tr> <td>31</td> <td>Tag records contain invalid data.</td> </tr> <tr> <td>32</td> <td>Duplicate Transaction_ID</td> </tr> </tbody> </table>	CODE	DESCRIPTION	00	The file was successfully validated.	10	File cannot be decrypted.	11	File cannot be uncompressed.	12	Badly formed XML file.	13	Duplicate File_ID	20	Summary data validation failed.	30	Transaction records contain invalid data.	31	Tag records contain invalid data.	32	Duplicate Transaction_ID	positiveInteger	SMALLINT, >0
CODE	DESCRIPTION																						
00	The file was successfully validated.																						
10	File cannot be decrypted.																						
11	File cannot be uncompressed.																						
12	Badly formed XML file.																						
13	Duplicate File_ID																						
20	Summary data validation failed.																						
30	Transaction records contain invalid data.																						
31	Tag records contain invalid data.																						
32	Duplicate Transaction_ID																						
SPECIAL_RULE_APPLIED	Notification if a special business rule was applied to the transaction. <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>SR167-Y</td> <td>Toll reduced to lower fare</td> </tr> <tr> <td>SR167-N</td> <td>Toll not changed.</td> </tr> </tbody> </table>	Value	Description	SR167-Y	Toll reduced to lower fare	SR167-N	Toll not changed.	string	VARCHAR(100)														
Value	Description																						
SR167-Y	Toll reduced to lower fare																						
SR167-N	Toll not changed.																						

Data Element Name	Comments	XML Data Type	TCAAS DB Data Type
TAG_AGENCY_ID	Agency ID associated with the Tag.	positiveInteger	SMALLINT, >0
TAG_CLASS	Class associated with the tag.	positiveInteger	SMALLINT, >0
TOTAL_TAG_COUNT_STATUS_0	Count of tags with status 0 in the Tag Status File.	unsignedLong	INT, >0
TOTAL_TAG_COUNT_STATUS_1	Count of tags with status 1 in the Tag Status File.	unsignedLong	INT, >0
TOTAL_TAG_COUNT_STATUS_2	Count of tags with status 2 in the Tag Status File.	unsignedLong	INT, >0
TOTAL_TAG_COUNT_STATUS_3	Count of tags with status 3 in the Tag Status File.	unsignedLong	INT, >0
TOTAL_TAG_COUNT_STATUS_4	Count of tags with status 4 in the Tag Status File.	unsignedLong	INT, >0
TOTAL_TAG_COUNT_STATUS_5	Count of tags with status 5 in the Tag Status File.	unsignedLong	INT, >0
TAG_HW_SERIAL_NUMBER	Hardware serial number associated with the tag. This is required for eGo Plus tags.	string	VARCHAR(32)
TAG_NUMBER	Serial number used to refer to the tag.	Long	INT, >0
TAG_PROTOCOL	Type of tag associated with number <u>VALUES</u> SEGO CVISN	string	VARCHAR(10)
TAG_REVENUE_TYPE	Revenue status of the Tag. (Applicable for all locations) <u>VALUES</u> REVENUE NONREVENUE	string	VARCHAR(10)
TAG_STATUS	<u>VALUE DESCRIPTION</u> 0 VALID 1 LOW BALANCE 2 INSUFFICIENT FUNDS 3 VERIFY – RESERVED FOR TRANSCORE USE 4 INVALID 5 LOST/STOLEN	positiveInteger	SMALLINT, >0
TOTAL_TAG_COUNT	Count of tags contained within the Tag status File	unsignedLong	INT, >0
TOLL	Amount of the Toll that is due to be deducted from a customer's account at the CSC.	positiveInteger	INT, >0
TRANSACTION_COUNT	Count of transactions contained within the Transaction file.	unsignedLong	INT, >0
TRANSACTION_ID	Server generated transaction ID, which uniquely identifies a transaction.	unsignedLong	INT, >0
TRANSACTION_SUM	Sum of Toll data elements for all transactions in the Transaction File.	positiveInteger	INT, >0
TRIP_SPEED	Speed of the vehicle as determined by the SR167 lane controller.	decimal	DECIMAL >0

Data Element Name	Comments	XML Data Type	TCAAS DB Data Type
TX_CLASS	Class associated with the transaction.	positiveInteger	SMALLINT, >0

Appendix A Tag Status File XML Schema

```

<xs:schema targetNamespace="http://localhost/SR167"
  xmlns:tsf="http://localhost/SR167"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:simpleType name="tagStatusType">
    <xs:restriction base="xs:positiveInteger">
      <xs:pattern value="[0-5]"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tagRevenueType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="REVENUE"/>
      <xs:enumeration value="NONREVENUE"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tagProtocolType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="SEGO"/>
      <xs:enumeration value="CVISN"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="TAG_STATUS_FILE_1.0">
    <xs:complexType>
      <xs:sequence minOccurs="1" maxOccurs="1">
        <xs:element name="HEADER">
          <xs:complexType>
            <xs:sequence minOccurs="1" maxOccurs="1">
              <xs:element name="FILE_DATE_TIME" type="xs:dateTime"/>
              <xs:element name="TOTAL_TAG_COUNT" type="xs:unsignedLong"/>
              <xs:element name="TOTAL_TAG_COUNT_STATUS_0" type="xs:unsignedLong"/>
              <xs:element name="TOTAL_TAG_COUNT_STATUS_1" type="xs:unsignedLong"/>
              <xs:element name="TOTAL_TAG_COUNT_STATUS_2" type="xs:unsignedLong"/>
              <xs:element name="TOTAL_TAG_COUNT_STATUS_3" type="xs:unsignedLong"/>
              <xs:element name="TOTAL_TAG_COUNT_STATUS_4" type="xs:unsignedLong"/>
              <xs:element name="TOTAL_TAG_COUNT_STATUS_5" type="xs:unsignedLong"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="DETAIL_DATA">
          <xs:complexType>
            <xs:sequence minOccurs="0" maxOccurs="unbounded">
              <xs:element name="TAG">
                <xs:complexType>
                  <xs:sequence minOccurs="1" maxOccurs="1">
                    <xs:element name="RECORD_ID" type="xs:unsignedLong"/>
                    <xs:element name="TAG_AGENCY_ID" type="xs:positiveInteger"/>
                    <xs:element name="TAG_PROTOCOL" type="tsf:tagProtocolType"/>
                    <xs:element name="TAG_REVENUE_TYPE" type="tsf:tagRevenueType"/>
                    <xs:element name="TAG_NUMBER" type="xs:long"/>
                    <xs:element name="TAG_HW_SERIAL_NUMBER" type="xs:string" minOccurs="0"/>
                    <xs:element name="TAG_CLASS" type="xs:positiveInteger"/>
                    <xs:element name="TAG_STATUS" type="tsf:tagStatusType"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="FOOTER">
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>

```


Appendix C Transfer Complete File XML Schema

```
<xs:schema targetNamespace="http://localhost/SR167"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:element name="TRANSFER_COMPLETE_1.0">
    <xs:complexType>
      <xs:sequence minOccurs="1" maxOccurs="1">
        <xs:element name="DETAIL_DATA">
          <xs:complexType>
            <xs:sequence minOccurs="0" maxOccurs="1">
              <xs:element name="FILE">
                <xs:complexType>
                  <xs:sequence minOccurs="1" maxOccurs="1">
                    <xs:element name="FILENAME" type="xs:string"/>
                    <xs:element name="FILETYPE" type="xs:string"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Appendix D Acknowledgement File XML Schema

```

<xs:schema targetNamespace="http://localhost/SR167"
  xmlns:tsf="http://localhost/SR167"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:simpleType name="ackReturnCode">
    <xs:restriction base="xs:integer">
      <xs:enumeration value="0"/>
      <xs:enumeration value="10"/>
      <xs:enumeration value="11"/>
      <xs:enumeration value="12"/>
      <xs:enumeration value="13"/>
      <xs:enumeration value="20"/>
      <xs:enumeration value="30"/>
      <xs:enumeration value="31"/>
      <xs:enumeration value="32"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="ACKNOWLEDGEMENT_1.0">
    <xs:complexType>
      <xs:sequence minOccurs="1" maxOccurs="1">
        <xs:element name="HEADER">
          <xs:complexType>
            <xs:sequence minOccurs="1" maxOccurs="1">
              <xs:element name="FILENAME" type="xs:string"/>
              <xs:element name="FILETYPE" type="xs:string"/>
              <xs:element name="RETURN_CODE" type="tsf:ackReturnCode"/>
              <xs:element name="NUM_RECORDS" type="xs:unsignedLong" minOccurs="0"/>
              <xs:element name="NUM_ISSUES" type="xs:unsignedLong" minOccurs="0"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="DETAIL_DATA">
          <xs:complexType>
            <xs:sequence minOccurs="0" maxOccurs="unbounded">
              <xs:element name="ERROR">
                <xs:complexType>
                  <xs:sequence minOccurs="1" maxOccurs="1">
                    <xs:element name="RECORD_ID" type="xs:unsignedLong"/>
                    <xs:element name="DESCRIPTION" type="xs:string"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="FOOTER">
          </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

This page intentionally left blank.

1 INTRODUCTION

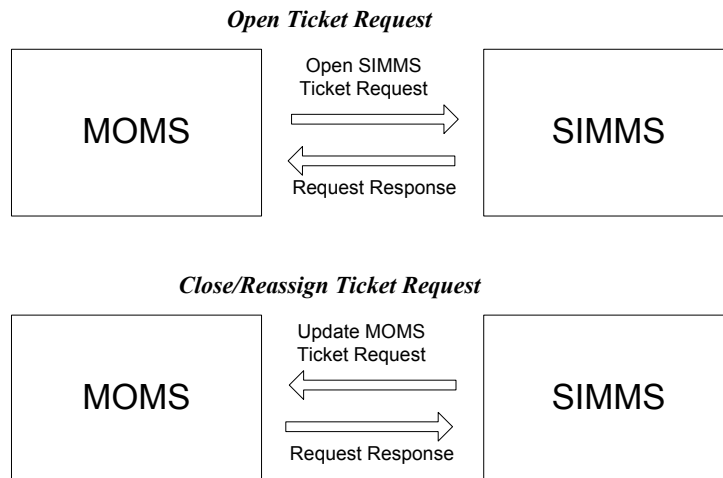
1.1 Purpose

This document describes the Web Services used by the FMAS and SIMMS to exchange ticketing information and requests.

1.2 Overview

The MOMS portion of the FMAS is responsible for evaluating the state of lane equipment and generating an open ticket request for SIMMS in the event that an issue is detected. When the issue has been resolved or it is determined that the ticket requires the involvement of ETC personnel, SIMMS sends a request to MOMS to take over responsibility of the ticket.

FMAS communicates with SIMMS by Web Service. This ICD defines the interfaces of the two Web Service.



2 SPECIFICATION

2.1 Type

This specification defines the API of Web Service used between FMAS and SIMMS.

2.2 Security

The data will be exchanged with no special security considerations.

2.3 Processing Guidelines

An open ticket request is sent to SIMMS with the MOMS ticket number. A SIMMS ticket is created and escalated by SIMMS. When the issue is resolved or needs reassignment to

ETC, SIMMS sends a request to MOMS for further action. The SIMMS request includes issue resolution notes for MOMS reporting purposes.

2.4 Web Service API

Method Summary	
void	<code>open_simms_ticket(java.lang.Integer moms_ticket_number, java.lang.Integer lane_id, java.lang.String lane_name, java.lang.String equipment_type, java.lang.String equipment_error)</code>
void	<code>update_moms_ticket(java.lang.Integer moms_ticket_number, java.lang.Integer simms_ticket_number, java.lang.String notes, java.lang.String action)</code>

Method Detail	
open_simms_ticket	
<pre>public void open_simms_ticket(java.lang.Integer moms_ticket_number, java.lang.Integer lane_id, java.lang.String lane_name, java.lang.String equipment_type, java.lang.String equipment_error) throws java.lang.Exception</pre>	
Parameters:	
<ul style="list-style-type: none"> moms_ticket_number - A unique identifier for this ticket in MOMS lane_id - A unique identifier for the Lane reporting an issue lane_name - The name of the lane reporting the issue equipment_type - The type of equipment for which an issue is detected equipment_error - The equipment error detected 	
Throws:	
<ul style="list-style-type: none"> java.lang.Exception - if SIMMS fails to open the ticket 	
update_moms_ticket	
<pre>public void update_moms_ticket(java.lang.Integer moms_ticket_number, java.lang.Integer simms_ticket_number, java.lang.String notes, java.lang.String action) throws java.lang.Exception</pre>	
Parameters:	
<ul style="list-style-type: none"> moms_ticket_number - A unique identifier for this ticket in MOMS simms_ticket_number - Unique ticket number from SIMMS notes - Notes as to the steps taken to correct the issue and/or steps required to correct the issue. action - Close/Reassign ticket, the value can only be "CLOSE" or "REASSIGN", it is not case sensitive. 	
Throws:	
<ul style="list-style-type: none"> java.lang.Exception - if MOMS fails to update the ticket 	

2.5 Sample Data

A sample open ticket in SIMMS request is shown below.

```
MOMSTicketNumber=1
LaneID=1
LaneName=167North-1
EquipmentType=AVI
```

EquipmentError= 3.8 - AVI Reader buffer full

A sample close ticket in MOMS request is shown below.

MOMSTicketNumber=1

SIMMSTicketNumber=1

Notes=Issue resolved at lane

Action=CLOSE

This page intentionally left blank.