Cascade Gateway Border Data Warehouse Upgrade & BIFA Integration Project
Report #1

Project Final Report

Whatcom Council of Governments
314 E. Champion Street, Bellingham, WA 98225
(360) 676-6974 • imtc@wcog.org • www.wcog.org

September 30, 2011

Table of Contents

Executive Summary

The Border Information Flow Architecture (BIFA) is a tool based on existing U.S. and Canadian National Intelligent Transportation Systems (ITS) Architectures, developed by the U.S. Federal Highway Administration (FHWA) and Transport Canada to ensure that technologies deployed at border crossings interact efficiently with one another. To facilitate the use of the BIFA, Transport Canada created a BIFA Pilot Project Funding Program to encourage recipients to undertake pilot projects that demonstrate the use of the BIFA to guide the implementation of technology at border crossings.

The Whatcom Council of Governments (WCOG) project was designed to build on an existing data archive and show how the BIFA can help plan cross-border ITS implementations. The project had three objectives.

Objective 1: Improve the binational Cascade Gateway Border Data Warehouse

Since the cross-border Cascade Gateway Border Data Warehouse is a prime example of a binational ITS project using multiple technologies owned by a variety of U.S. and Canadian agencies, it was used at the sample project for developing a BIFA project architecture.

Improving the original archive

The previous border data warehouse had several issues identified by regional stakeholders which affected functionality and needed improvement. Through stakeholder feedback and the Project Advisory Team, a list of limitations in the prior system was identified, along with seven operational needs to address in the new system:

1. The warehouse needs to maintain existing features (including but not limited to port and detector data, email automated reports, downloadable files, and custom queries).
2. The warehouse needs to present data in a clear, concise manner that is easy for inexperienced end-users to find what they are looking for.
3. The warehouse needs to provide data in downloadable and exchangeable formats that are easy to understand and capable of being used on multiple platforms.
4. The warehouse needs to quickly and efficiently pull data from the archive.
5. The warehouse (website and data archive) needs to be developed in such a way as to make it simple to modify and be alerted of errors.
6. The warehouse needs to be designed in a way that administrators can monitor the health of the warehouse and be alerted of errors.
7. The warehouse needs to be built for future expansion and increased archiving requirements.

Systems Engineering documentation was developed as well as a project architecture using the BIFA template. Using these materials, WCOG hired IBI Group to develop the new archive and corresponding website. The website and data set are available at: www.CascadeGatewayData.com.

Incorporating new data

In addition to developing the new website, IBI Group was hired to examine the possibility of integrating additional data sources into the archive:

1. U.S. Bureau of Transportation Statistics website archive data
2. B.C. weigh-in-motion detector data
3. WA State weigh-in-motion detector data
4. CVISN data on I-5 northbound
5. GPS fleet data (purchased from vendors by WA State)
6. Booth status data

Of the data sets analyzed, work continued on three datasets: the BTS commodity data, the WA State weigh-in-motion detector data, and the GPS fleet data. The B.C. weigh-in-motion detector data was determined to be of too poor quality to be worth the expense. The CVISN data would have taken longer than the project timeline to provide power and communications for data transfer. The booth status data is still dependent on pending decisions within U.S. Customs & Border Protection and remains an interest of IMTC stakeholders.

The BTS commodity data are now available as a tab on the website. GPS fleet data previously provided by WSDOT are archived in the database but not yet queriable online, pending future data transfers and a determination of how best to use the data in the context of transportation planning. Lastly, work continues to link the WSDOT weigh-in-motion detector to the archive. Equipment has been purchased but there still needs to be improvements to communication for wireless transfers of the data to the system.

Objective 2: Use the BIFA template to develop a project architecture that will be compatible with both U.S. and Canadian regional ITS architectures

WCOG used the BIFA template of Turbo Architecture as the starting point in developing a project ITS architecture that would be compatible not only with the Whatcom Regional ITS Architecture, but with B.C.'s ITS strategic plan.

A draft of the project architecture was reviewed by Consystec and R.C. Ice and Associates for issues and recommendations, and revised. The final version is available as **Border Data Warehouse Upgrade & BIFA Integration Project Report #2: Project ITS Architecture.**

The Turbo Architecture project file was also converted to U.S. and Canadian Turbo-compatible versions so that B.C. architecture developers can integrate the project seamlessly into their versions of their regional architecture.

The entire architecture development process was detailed in a log for the benefit of other agencies developing similar cross-border ITS projects to highlight methodology, what worked, and how issues were resolved. The log is available as part of: **Border Data Warehouse Upgrade & BIFA Integration Project Report #3: Recommendations for BIFA and Architecture Developers**.

Objective 3: Provide guidance for other regions considering the BIFA for their cross-border ITS projects, and recommendations for future BIFA development

An original deliverable envisioned for this project was a user's manual for regions developing similar binational ITS projects. However as WCOG used the BIFA template for the architecture process, it was quickly determined that, for numerous reasons, using the template was harder than starting an architecture from scratch.

Instead of a user's manual, therefore, WCOG developed a more robust recommendations report for both BIFA developers and for agencies interested in creating a binational project architecture. The recommendations report is available as: **Border Data Warehouse Upgrade & BIFA Integration Project Report #3: Recommendations for BIFA and Architecture Developers**.

Benefits of using BIFA

The greatest benefit of the BIFA was seen at the initial stages of the project and included:

- Developing a BIFA-based project architecture was a valuable exercise in identifying project stakeholders
- Mapping data-sharing connections made it easier to describe the project with stakeholders and define elements needed to make those connections work
- It helped clarify which data-sharing agreements would be useful long-term
- It mapped connections between the U.S. and Canadian services.

Challenges of using BIFA

Although there were benefits using the BIFA for this project architecture, numerous challenges affected the overall process for WCOG. Challenges included:

1. BIFA usability
2. Turbo architecture software
3. Using BIFA for a project
4. Incompatibilities with U.S./Canadian architectures
5. Unique BIFA components are obsolete

Challenges of ITS Architectures in general

There were also challenges to the ITS architecture process in general, which were not specific to BIFA but just as critical in this project. To better identify how many of these challenges were unique to this project and how many were shared by other regional ITS architecture maintainers, a series of interviews were conducted and results compiled. Challenges identified through this interview process included:

1. Inaccessible language
2. Repetitive structure
3. Architecture is too specific for planners but not specific enough for engineers
4. Architecture is not being used as intended

Recommendations

Recommendations developed for the BIFA itself include:

1. Re-define BIFA as a template for guiding cross-border ITS projects with recommendations for strategies, not as a separate architecture.

2. Do not develop cross-border ITS projects using the Turbo files unless a version of Turbo can be made compatible with both U.S. and Canadian architectures.
3. Use regional architectures in lieu of developing a separate BIFA.

Recommendations for national ITS architecture changes were also developed and outlined in the report, as well as recommendations for agencies undergoing a similar process.

More details can be found in: **Border Data Warehouse Upgrade & BIFA Integration Project Report #3: Recommendations for BIFA and Architecture Developers**.

1. Introduction

The Border Information Flow Architecture (BIFA) was developed jointly by the U.S. Federal Highway Administration (FHWA) and Transport Canada to ensure that technologies deployed at border crossings interact efficiently with one another. Based on existing U.S. and Canadian National Architectures, the tool is designed to be used for Intelligent Transportation Systems (ITS) project implementation at or near border crossings, as a template for border area regional ITS architecture updates, and to support the transportation planning process.

The BIFA emerged from discussions regarding cross-border ITS coordination at the binational Transportation Border Working Group (TBWG) forum. Federal, provincial, state, and local representatives from transportation and inspection agencies provided feedback which led to the development of the BIFA.

The BIFA is available online at www.iteris.com/itsarch/bifa. A full report is available as well as Turbo Architecture files for customizing the BIFA for a specific region or project.

To facilitate the use of the BIFA, Transport Canada created a BIFA Pilot Project Funding Program to encourage recipients to undertake pilot projects that demonstrate the use of the BIFA to guide the implementation of technology at border crossings.

1.1 Border Data Warehouse Upgrade & BIFA Integration Project

The Whatcom Council of Governments (WCOG) received funding through Transport Canada's BIFA Pilot Project Funding Program in 2010, matched with funding from WA State Department of Transportation (WSDOT) and FHWA, to support ongoing public and private planning operations of the Cascade Gateway border crossing system[1], and to advance the BIFA itself. The project was designed to build on an existing data archive and show how the BIFA can help plan cross-border ITS deployments. The project had three objectives:

1. Improve the binational Cascade Gateway Border Data Warehouse
2. Use the BIFA template to develop a project architecture that will be compatible with both U.S. and Canadian regional ITS architectures
3. Provide guidelines for other regions considering the BIFA for their cross-border ITS projects, and recommendations for future BIFA development.

This report summarizes activities undertaken and deliverables produced for this project. More in-depth reporting of the project architecture can be found in **Border Data Warehouse Upgrade & BIFA Integration Project Report #2: Project ITS Architecture.** A detailed list of recommendations is available in **Border Data Warehouse Upgrade & BIFA Integration Project Report #3: Recommendations for BIFA and Architecture Developers**.

2. Border Data Warehouse Improvements

The first task of the project was to improve the binational Cascade Gateway Border Data Warehouse, using the BIFA template as a guideline for the project.

[1] The Cascade Gateway is the system of four border crossings between the Lower Mainland of British Columbia, and Whatcom County in Washington State. The crossings include Peace Arch/Douglas, Pacific Highway, Lynden/Aldergrove, and Sumas/Abbotsford-Huntingdon ports-of-entry.

The project was chosen as a prime example for a BIFA project because it involves ITS technology spanning the border, and multiple U.S. and Canadian agencies working together to share data to improve the border crossing as a whole.

2.1 Existing Advanced Traveler Information Systems

WSDOT maintains a border Advanced Traveler Information System (ATIS) which provides estimated delay for northbound passenger vehicle travelers crossing the U.S. – Canada border. The initial system developed prior to 2006 included Peace Arch and Pacific Highway ports-of-entry, and was designed to optimize usage of the two crossings by spreading demand across both ports, which are less than a mile apart. The ATIS system included variable message signs (VMS) showing comparative wait times, highway advisory radio (HAR) reports of wait times, and a website[2] with both wait times and camera images of current conditions at the crossings.

The system proved to be effective in evening out average wait times between both crossings. In 2008 WSDOT expanded the system to include the other two Cascade Gateway ports-of-entry, Lynden/Aldergrove and Sumas/Huntingdon. Although these crossings are further from the primary passenger vehicle crossings, they may still provide an alternative if travelers are alerted early enough..

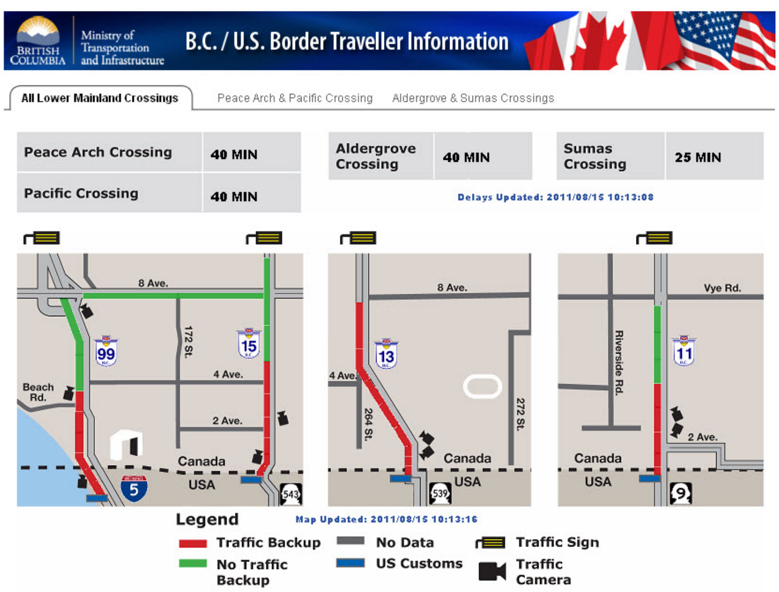


Figure 1: BCMOT's Border Traveller Information System Website (Aug. 2011)

In addition to providing wait times for passenger vehicles, the WSDOT system also calculates estimated wait times for commercial vehicles northbound at Pacific Highway port-of-entry.

Southbound, a similar system was built by B.C. Ministry of Transportation (BCMOT) in 2000 to provide estimated wait times for Peace Arch/Douglas and Pacific Highway ports-of-entry. The system included VMS signs and a website[3] with camera images and congestion map as well.

The southbound system was altered slightly in 2008 to provide data to the anti-idling system incorporated at the Peace Arch port-of-entry, which includes traffic lights which release queuing traffic in platoons, allowing travelers to turn off their engines when the lights are red. This project was part of a Greening the Border provincial initiative.

The southbound system was expanded to the Lynden/Aldergrove and Sumas/Huntingdon crossings in 2011 and include both VMS and hybrid message signs showing wait times at all four Cascade Gateway ports-of-entry.

---

2 http://www.wsdot.wa.gov/Traffic/Border/

3 http://www.th.gov.bc.ca/ATIS/index.htm

2.2 CascadeGatewayData.com

Prior to 2006, data from the WSDOT and BCMOT systems was not archived. Participants from the International Mobility & Trade Corridor Project (IMTC)[4] identified a need to preserve this valuable data and archive historic wait times, and in 2006 developed www.CascadeGatewayData.com as a border data warehouse.

The original data warehouse archived all data from the BCMOT and WSDOT border ATIS systems in five minute increments, and then availed the stored data in an online database. The database was accessible to the public and provided traffic volume data, delay, arrival rates, and other pertinent data by crossing and direction. There were also custom query functions and the ability to view data by individual loop detector.

Figure 2: Original Cascade Gateway Border Data Warehouse (April 2007)

Back-end functionalities of the archive included an email notification system whenever border wait times exceeded a specific threshold, and a calculation for the website manager to determine a percentage of data packets received every day.

The original system was designed to use FTP protocols established through an agreed-upon inter-agency XML schema. The archive was on a UNIX server using Ruby on Rails backed by a PostgreSQL database, with the website presented by Apache.

The site went live in 2007 and ran until January 1, 2010, at which time the data continued to be stored but the website itself went down.

One of the challenges of the original system was that its design, using effective but relatively-unknown software systems like Ruby, was difficult to maintain. Finding skilled developers proved to be problematic, and by 2010 it was clear the original design of the archive lacked features needed to optimally store an increasing data set in a way that could be efficiently accessed over the internet.

Numerous limitations of the system made the archive difficult to query and inaccessible by many users (a full description of all limitations can be found in **Appendix A: Concept of Operations**). It was determined that a full upgrade of the archive and website would be required to keep needed functionality of the data warehouse.

[4] The International Mobility & Trade Corridor project is a U.S. – Canadian coalition of government and business entities that identifies and promotes improvements to mobility and security for the Cascade Gateway border crossings. IMTC participants identified the need for, and advised, this project. More information about IMTC can be found at: www.wcog.org/imtc.

2.3 Identifying Improvements

A project advisory team consisting of representatives from both BCMOT, WSDOT, Transport Canada, and both U.S. and Canadian inspection agencies was established to define the needed improvements to the border data warehouse. Through this process, fixes were identified, as well as opportunities to improve the quality and scope of data available through the warehouse.

In addition to the project advisory team suggestions, feedback was also collected from other stakeholders and warehouse users through an online survey. Feedback received for improving the project was detailed in a report attached as **Appendix B: Feedback Report**.

Based on the feedback received, the following operational needs were identified:

1. The warehouse needs to maintain existing features (including but not limited to port and detector data, email automated reports, downloadable files, and custom queries).
2. The warehouse needs to present data in a clear, concise manner that is easy for inexperienced end-users to find what they are looking for.
3. The warehouse needs to provide data in downloadable and exchangeable formats that are easy to understand and capable of being used on multiple platforms.
4. The warehouse needs to quickly and efficiently pull data from the archive.
5. The warehouse (website and data archive) needs to be developed in such a way as to make it simple to modify and be alerted of errors.
6. The warehouse needs to be designed in a way that administrators can monitor the health of the warehouse and be alerted of errors.
7. The warehouse needs to be built for future expansion and increased archiving requirements.

2.4 System Engineering Documentation

Because this project was paid for in part by FHWA funds, a full systems engineering analysis was completed to document how the warehouse upgrade needs were identified and traced to specific improvements[5]. Each operational need listed above can be traced back to a specific limitation identified by stakeholders and forward to a specific requirement (see **Appendix C: System Requirements Traceability Matrix**).

The Project Management Plan (see **Appendix D: Project Management Plan**) defined the scope of work, tasks, and deliverables for the project as well as the resources, budget, and estimated schedule.

Measuring the effectiveness of the project in terms of how identified needs have been specifically addressed in the new system is detailed in **Appendix E: System Validation Plan**. The final systems engineering document is **Appendix F: System Verification Plan**.

2.5 Selection of Project Developer

WCOG developed a request for proposals (RFP) to complete work on the border data warehouse. Selection criteria were based on understanding of the project, a methodology and approach that would prevent the issues experienced in the original design; a statement regarding long-term

[5] More information about Systems Engineering is available in FHWA's online guidebook: http://www.fhwa.dot.gov/cadiv/segb/index.htm

maintenance and why their chosen programming platform should be preferred over other approaches; qualifications and experience of the project team, and cost.

IBI Group was selected to re-design the website and data archive. They were also selected to work with stakeholder agencies to investigate the possibility of incorporating the additional data sets illustrated on the original schematic.

2.6 Warehouse Design and Functionality

To address many of the concerns listed in the Concept of Operations, the consultants chose to develop the new archive and website with Microsoft .NET 4.0 using a SQL Server for the database. Data is imported into the system through FTP and HTML protocols.

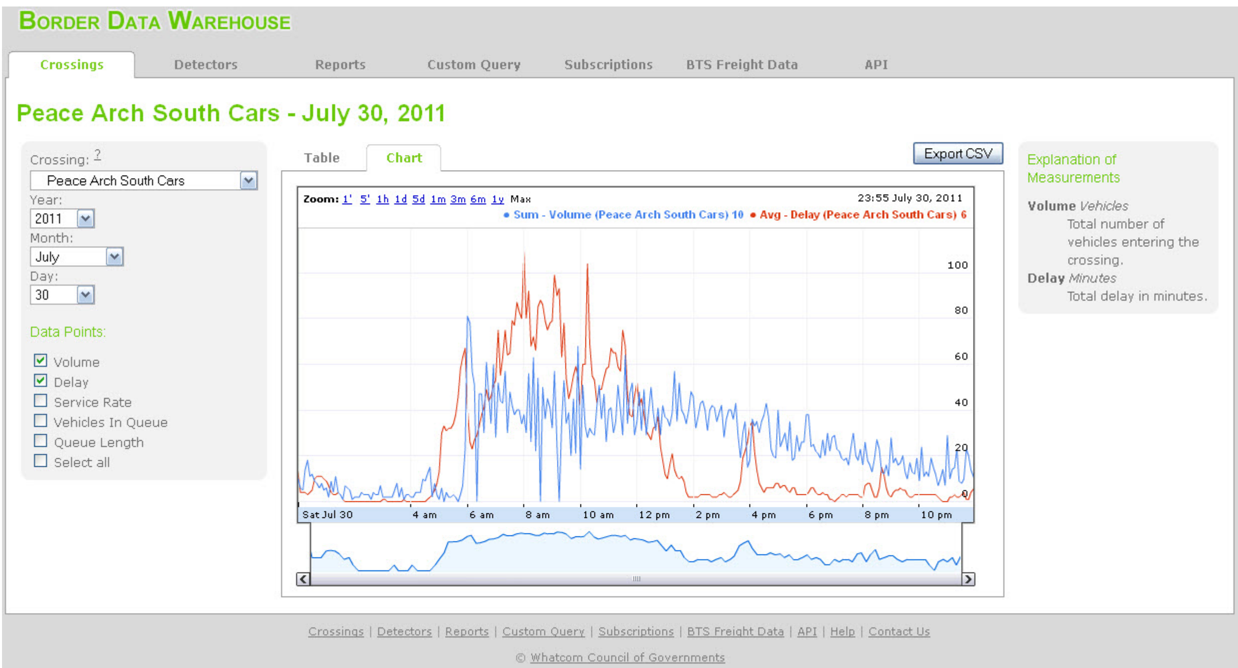


Figure 3: Data output from the new Border Data Warehouse at www.CascadeGatewayData.com

The new website now performs quickly and efficiently; presents data in a clear and accessible manner; is designed for better long-term maintenance; and offers numerous new features and enhanced functionality, including:

- A new server purchased as part of the project to house the increasing amounts of data
- Elimination of data duplicates and a new script to prevent duplications from occurring
- A new feature to monitor the availability of data packets for a given date and crossing or detector for better accuracy in reporting
- Full administrative functionality for site managers
- Interactive geocoded mapping capability
- Ability to save custom queries as packaged reports
- Google charting abilities

Improvements tracked to specific requirements and needs are mapped in **Appendix C: System Requirements Traceability Matrix**.

2.8 Review of Additional Datasets

In addition to developing the Border Data Warehouse and accompanying website, IBI Group was also hired to evaluate additional datasets to consider including in the archive.

Bureau of Transportation Statistics' Transborder Surface Freight Database

The Research and Innovative Technology Administration Bureau of Transportation Statistics (BTS) North American Transborder Freight Database is an online resource for many regional border stakeholders, referenced regularly to show commodity types, values, and weights that cross through the Cascade Gateway. Since this resource is regularly used, several stakeholders saw benefit in having a feed from the existing BTS website to the Border Data Warehouse to provide easier querying and make the website a "one stop shop" for Cascade Gateway border data.

Discussions with representatives from BTS clarified that internal BTS security protocols did not allow for FTP access to data, and reports would have to be manually compiled on a monthly basis. However project website developers were able to develop a query tool that used the existing BTS website and display data results from the source archive at:

http://www.bts.gov/programs/international/transborder/TBDR_QAPC07.html.

The mirrored data is displayed in the Border Data Warehouse format and has all the functionality of the rest of the warehouse. However data are not stored separately in the Cascade Gateway archive, and the results are dependent on the BTS query tool. The benefit of this is that data are not being stored in duplicate in the Cascade Gateway Border Data Warehouse; however the downside is that if any changes occur to the BTS URL or if their website goes down, it will affect the functionality of the BTS component of this archive.

CVISN Data

Commercial Vehicle Information and Systems Networks (CVISN) antenna data are available for the Interstate 5 corridor from the Port of Seattle to the border crossing at Pacific Highway. These detectors were originally installed as part of a pilot project with the Advanced Technology Branch of WSDOT, and provide a unique identifier for each truck with a transponder and a date and time for the read of each transponder.

Communication between these transponders and an FTP or other connection does not currently exist, and would need to be added in order for data to be automatically transferred.

Given that this data set is limited only to trucks with specific transponders on board; data are only available for the U.S. side of the I-5 corridor; and specific delay data at the border would be difficult to calculate given the locations of the detectors at the border, it was determined that, while possible, project time limitations prevented incorporating this data set at this time.

Regional GPS Fleet Data

GPS data collected from GPS system vendors and shared with WSDOT included geocoded movements of approximately 6,000 trucks through the State of Washington. The data include a unique identifier for a truck; a date and time stamp for each read; a direction of travel; latitude and longitude data; and whether the vehicle is active or parked.

The developers were asked to construct a data table that would store vendor-purchased GPS fleet data which could be mined at a later date for such information as truck density, truck trips, origin and destination, and trip duration.

Since future GPS data sets would need to be purchased, and the output from such data sets has yet to be specifically defined, the developers created a table with the existing 6,000 records as a starting point and provided a mechanism by which the archive could automatically be updated at a later date.

Until the parameters of what should be displayed from the database is better clarified, the data are stored in the warehouse but not accessible to the public.

Inspection Booth Status Data

Because both northbound and southbound ATIS systems depend on embedded loop detectors to provide data to the systems regarding service rate, queue length, and vehicle counts, the systems require a relatively constant physical landscape at the border.

Since initial development, northbound and southbound ATIS system exclude NEXUS data from wait time calculations by treating data from loops in that lane separately. However this distinction between NEXUS and regular traffic is getting more complicated since recent changes at border crossings mean that dynamic lane changes occur, leading to possible discrepancies in the ATIS systems.

To better reflect real-time booth status changes, the northbound and southbound ATIS systems would need data feeds from the inspection agencies themselves as to the changing status of lanes.

This remains a goal of both ATIS systems. And it is hoped such data may be archived in the warehouse. However the inspection agencies are currently exploring ways to make such data available at this time. Work therefore could not be completed on this in the time frame of this project.

Weigh-in-Motion Detector Data

Weigh-in-motion (WIM) detectors use scales embedded in the roadway to weigh commercial vehicles and determine some features of the vehicle including vehicle type (by axle-count) and speed.

Southbound detectors

In 2006 WIM detectors were installed on B.C. Highways 15, 13, and 11 as part of the National Roadside Survey conducted by Transport Canada. After this survey effort ownership of the detectors transferred to BCMOT.

This portion of the project looked at whether data from these detectors could be automatically transferred from the detectors to the border data warehouse archive, thereby providing vehicle type and vehicle count data comparable to other data collection systems in the region.

However upon further investigation it was discovered that the locations of the detectors do not provide wireless communications effectively. All data must be manually downloaded at the WIM detector sites. In addition, the data are provided in a proprietary format and not easily converted into a format that could be incorporated with the existing warehouse schema.

Lastly, the quality of the data were brought into question. The detector vehicle type measurements are not very accurate and much of the data are considered unusable. Therefore it was determined not to explore this option further.

Northbound detectors

A WIM detector on Interstate 5 prior to Exit 275 was installed as part of the CVISN system in 2004. CVISN's original intent was to use this as a virtual WIM site. Although they had several locations in various forms of progress they eventually ran low on funds and the project was scrapped.

As part of this project, WCOG paid for hardware improvements to the Piezo weigh scale to allow for more accurate data. Next steps in this project include establishing a way to automate data transfers and extract the files through FTP into the archive.

The objective of hooking up this detector is that it can provide accurate volume counts and vehicle types in both directions on I-5, which can serve as both a comparable data set to other volume counters, as well as specific commercial vehicle data currently not available (i.e. empty vs. loaded trucks).

2.9 Future improvements

The Cascade Gateway Border Data Warehouse has been completed as per the original scope of the project. However there are numerous additions that can be made to improve the system.

Additional data

Although the project was unable to incorporate booth status data within the time frame of this project it is hoped that these data will be made available in the near future, and can be archived.

In addition, explorations of the availability and usability of the GPS fleet data will continue.

Lastly, the WIM detector data has yet to be fully incorporated into the site and work will continue to make this happen.

New features

Now that the new archive has been made available to stakeholders for several months, feedback suggesting additional improvements have been received with the hopes of incorporating changes in the near future. Some of these future improvements include:

- The ability to view detector maps on the custom query page for easier detector-level query building
- Ability to save custom query URLs as a unique address so queries can be emailed from the original website user to others for discussion
- Improvements to the email subscription functionality which would allow for administrators to view which reports are subscribed to (for system performance measurement), the ability to assign a report to multiple crossings at the same time, etc.
- Improvements to the BTS data component including providing aggregated data and other features not available on the original site
- Better customizable layouts of pages (for the administrator) to provide for notes, comments, and warnings on various web pages.

Maintenance

WCOG has set aside $3,000 per year for the next five years ($15,000 total) toward maintaining the website. These funds will cover administration, hosting, server maintenance, and backup services for the site.

Funding for more in-depth maintenance, including more serious errors or complications with the archive, is being sought separately.

3. BIFA-based Project Architecture

Using the BIFA template, a project architecture was developed to represent the scope of data shared between agencies as part of the project; the types of ITS services being used in the project; data sharing agreements; standards incorporated into elements of the project; and the relationships of stakeholders. The architecture was initially developed using a schematic of how new options for exchanging information with regional stakeholders might occur. Using the BIFA template, WCOG began work on mapping the architecture using the Turbo Architecture software available on the FHWA website[6].

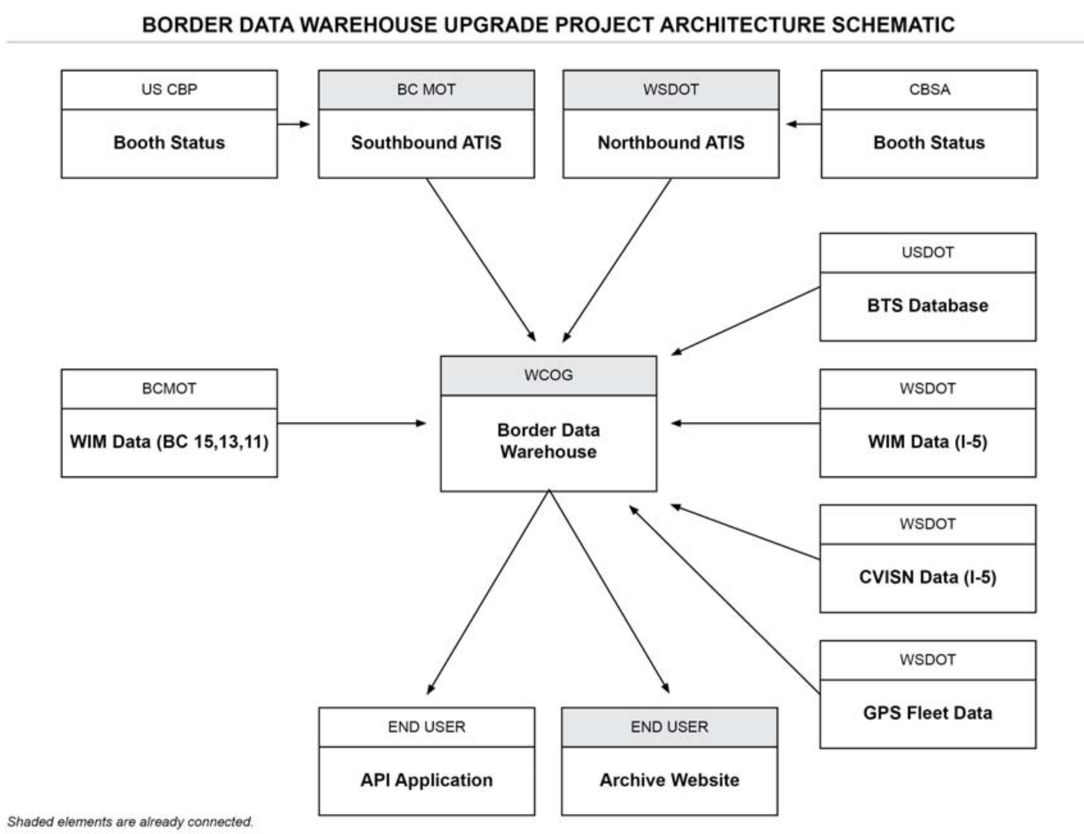


Figure 4: Original project schematic and basis for architecture

[6] http://www.iteris.com/itsarch/html/turbo/turbomain.htm

3.1 Initial architecture development

The initial project architecture was developed in-house by WCOG staff. Staff had previously been trained in ITS architecture management through Regional ITS Architecture workshops provided by FHWA, and training courses on using Turbo Architecture, a software application specifically developed to design and maintain regional and project architectures.

The **Border Information Flow Architecture Final Report** (National Architecture Team, January 2006) was referenced prior to beginning the project, and elements related to this specific project highlighted. Notes on the BIFA final report were recorded in the BIFA Log (see **Border Data Warehouse Upgrade & BIFA Integration Project Report #3: Recommendations for BIFA and Architecture Developers)**.

The BIFA Turbo Architecture template was determined to be the best starting point given the lack of prior experience from other users. Therefore the U.S. ITS architecture template file was chosen for this project (however there is also a separate BIFA template for the Canadian National ITS Architecture available).

3.2 Issues with the BIFA

Numerous issues were raised with the BIFA report in attempting to translate the effectiveness of the architecture in the context of a small project scope.

1. **BIFA usability** – the biggest issue was that the BIFA architecture serves as both a complete architecture on its own, for border areas that do not have regional ITS architectures already in place, as well as a template for regions that *do* have existing architectures to use. By trying to serve both purposes, the usability of the BIFA diminishes and serves neither role well.
2. **Turbo Architecture** – The Turbo Architecture took developed by FHWA and Transport Canada is the encouraged mechanism for agencies to create and maintain regional and project architectures. However there are numerous challenges with this tool. It is too large for practical use in a project architecture, requires too much training, and files are not transferrable between U.S. and Canadian versions of the Turbo Architecture.
3. **Using BIFA for a project** – Too much customization is needed of the BIFA to use effectively for planning a specific ITS project at the border. In addition, certain bugs within the Turbo tool make it very difficult to develop a project architecture separate from e regional architecture.
4. **Incompatibilities with U.S. and Canadian architectures** – Although the original purpose for BIFA was to create a mechanism for integrating cross-border ITS projects seamlessly, this is not possible in the current version of Turbo without a great deal of work and paid assistance from the developer of the tool.
5. **BIFA-unique components obsolete** – The BIFA-unique elements developed to describe specific information flows at border crossings have been incorporated into the Canadian National Architecture Version 2.0 and are being discussed as an addition in the next update to the U.S. National ITS Architecture.

All of these concerns are discussed in greater details in the **Border Data Warehouse Upgrade & BIFA Integration Project Report #3: Recommendations for BIFA and Architecture Developers.**

3.3 Using Turbo Architecture

The project architecture was developed on the BIFA template of the U.S. national ITS architecture in Turbo Architecture. The user's experience in developing the architecture in this manner was recorded in the BIFA Log, available as part of **Border Data Warehouse Upgrade & BIFA Integration Project Report #3: Recommendations for BIFA and Architecture Developers.**

The Turbo Architecture file was used to develop project schematics during the course of the project development. These schematics were shared with the developers to better illustrate the intended flow of information between stakeholder agencies.

3.4 Validation of the Project Architecture

Because no project architecture using the BIFA previously existed, the draft architecture was sent to two renowned developers and ITS project experts, Bruce S. Eisenhart from ConSysTec Corporation, and Ronald C. Ice from R.C. Ice and Associates.

Both consultants reviewed the draft architecture and responded with recommended approaches and changes. Changes were made and discussed in detail, and the architecture sent back for final review.

On the whole the reviewers found the project architecture to be an accurate representation of the project that adhered to the objectives of the BIFA.

3.5 Modifications to Develop a Canadian File

Several modifications to the final Turbo file were made by the developer Ron Ice that could not be done by WCOG. One of the modifications addressed a "bug" of sorts in the Turbo file that, until then, occurred when a project architecture is used without an overarching regional architecture to define it.

Another modification was to convert the file into a format compatible with the Canadian National ITS Architecture. It was hoped this could be something completed by WCOG, but the amount of programming involved in developing a true Turbo file that spans the border is prohibitive. This issue is worth further discussion by BIFA stakeholders and is discussed in more detail in the recommendations report.

3.6 Finalizing the Project Architecture Final Report

The final report was written by WCOG at the completion of the upgrade. It is written in a format compatible with the U.S. National Architecture standards. A full copy of the Project Architecture is available as **Border Data Warehouse Upgrade & BIFA Integration Project Report #2: Project ITS Architecture.**

4. Guidelines for Other Users

An original project objective established in the scope of work was to develop guidelines for agencies developing similar cross-border ITS projects looking to develop effective ITS architectures to assist in the project scoping and integration with regional systems.

However throughout the course of this project it was discovered that numerous challenges face an individual agency in using the BIFA document or template as guidance in developing their binational ITS projects. Instead of writing a user's guide, therefore, it was determined that it would be more beneficial to focus on developing recommendations for both BIFA and ITS Architecture developers as well as recommendations for those agencies interested in making architectures for binational projects, not using the BIFA.

Generalized recommendations for other cross-border project managers, however, include the following:

- Define the schematic of data sharing and entities at the beginning of the project
- Use the U.S. required systems engineering process to define and manage the project
- Do not spend time developing a BIFA project architecture.

More details are available in: **Border Data Warehouse Upgrade & BIFA Integration Project Report #3: Recommendations for BIFA and Architecture Developers.**

5. Recommendations for BIFA and ITS Architectures

Based on the experience of this project, WCOG has developed a series of recommendations focused primarily on the BIFA tool but also some questions regarding the ITS Architecture process in general, and how it can better serve the original purpose of helping coordinate and integrate regional ITS investments.

Recommendations are detailed in **Border Data Warehouse Upgrade & BIFA Integration Project Report #4: Recommendations for BIFA and Architecture Developers**

6. For More Information

The Cascade Gateway Border Data Warehouse is available online at:
www.CascadeGatewayData.com.

Project reports, appendices, and all documentation are available on the project website which can be found at www.wcog.org/border.

More information is available by contacting:

Whatcom Council of Governments
314 E. Champion Street
Bellingham, WA 98225
(360) 676-6974

imtc@wcog.org.

Further Reading

1. **Border Information Flow Architecture Final Report**, January 2006, National ITS Architecture Team: http://www.ronice.com/bifa/bifadocument01-19-06.pdf
2. **Border Data Warehouse Upgrade & BIFA Integration Project Report #2: Project ITS Architecture,** September 2011, Whatcom Council of Governments
3. **Border Data Warehouse Upgrade & BIFA Integration Project Report #3: Recommendations for BIFA and Architecture Developers,** September 2011, Whatcom Council of Governments

Appendix A: Concept of Operations for the Cascade Gateway Border Data Warehouse

1. Introduction

The Cascade Gateway Border Data Warehouse is a U.S. – Canadian data storage server and corresponding database that archives real-time traffic data relating to the Cascade Gateway system of border crossings. The original server and website were developed in 2002, funded by Transport Canada, B.C. Ministry of Transportation (BCMOT), and with an in-kind contribution from WA State Department of Transportation (WSDOT). The warehouse is managed by the Whatcom Council of Governments (WCOG), the Metropolitan Planning Organization (MPO) in Bellingham, Washington and lead agency for the International Mobility & Trade Corridor Project (IMTC).

The International Mobility & Trade Corridor Project

IMTC is a U.S. – Canadian coalition of government and business entities that identifies and promotes improvements to mobility and security for the four border crossings that make up the Cascade Gateway. The goals of the IMTC project are to facilitate a forum for ongoing communication between agencies responsible for regional, cross-border transportation, safety, and security; coordinate planning of the Cascade Gateway as a transportation and inspection system rather than as individual border crossings; improve and distribute traffic data and information; and identify and pursue improvements to infrastructure, operations, and information technology.

The Cascade Gateway

The Cascade Gateway refers to the four land ports-of-entry between Whatcom County in Washington State and the Lower Mainland of British Columbia. The four crossings include Peace Arch/Douglas (accessed by Interstate 5/B.C. Highway 99), Pacific Highway (State Route 543/B.C. Highway 15), Lynden/Aldergrove (State Route 539/B.C. Highway 13), and Sumas/Huntingdon (State Route 9/B.C. Highway 11).

Advanced Traveler Information Systems

As a critical goal of the IMTC project, data collection and dissemination is vital to regional planning agencies, operators, and the traveling public for better management of the cross-border transportation system. Both U.S. and Canadian federal governments have invested in improving the type and quality of data at border crossings to better understand the dynamics of traffic flow and to measure effectiveness of other regional improvements. Advanced Traveler Information Systems (ATIS) at border crossings have been a priority of the IMTC coalition since its foundation, and several binational funding initiatives have helped to develop border wait time systems in both directions to capture data and inform travelers of expected delays.

BCMOT and WSDOT both operate independent ATIS systems in the Cascade Gateway. Each system relies on equipment in the road (loop detectors) to detect vehicles. The systems calculate traffic volumes and arrival rates, and use these figures to determine an estimated wait time, which can then be compared to wait times at other nearby crossings.

Northbound, WSDOT provides passenger vehicle wait times at all four border crossings, and truck wait times at the Pacific Highway port-of-entry. Southbound, BCMOT provides passenger vehicle wait times at Peace Arch/Douglas and Pacific Highway ports-of-entry only, although data are collected on commercial vehicle movements at Pacific Highway, and the system will be expanded in 2010/2011 to include the other two ports.

Data are shared via variable message signs on major regional routes and on agency websites. Live WSDOT data can be found on the website at: http://www.wsdot.wa.gov/Traffic/Border/ and live BCMOT data can be found at: http://www.th.gov.bc.ca/ATIS/index.htm.

CascadeGatewayData.com

Prior to 2006, data from the WSDOT and BCMOT systems was not archived. IMTC participants identified a need to preserve this valuable data and archive historic wait times, and in 2006 developed the www.CascadeGatewayData.com website and corresponding border data warehouse. This warehouse archives all streaming data in five minute increments, and then displays the stored data in an online database. The database is accessible to the public and provides traffic volume data, delay, arrival rates, and other pertinent data by crossing and direction. There are also custom query functions and the ability to view data by individual road loop detector.

There are also back-end functionalities to the website including an email notification system whenever border wait times exceed a specific threshold.

Using FTP protocols established on an agreed-upon inter-agency XML schema, BCMOT and WSDOT data are transferred hourly to the warehouse server and then displayed on the website.

The existing site is managed by WCOG and the server stored at CSS Communications in Bellingham, Washington. WCOG owns the server.

The warehouse is unique across the U.S. – Canada border and held as an example of true binational data sharing and planning

BIFA Pilot Program

In December, 2009, WCOG applied for funding under the Transport Canada Border Information Flow Architecture (BIFA) Pilot Project Program to improve functionality and user access to the existing warehouse; increase the amount and types of data available to all users; document how a binational project can use BIFA Intelligent Transportation System (ITS) architecture to improve cross-border ITS planning; and to develop a manual for BIFA integration usable by other regions looking to strategically assess and respond to data sharing and system integration opportunities.

WCOG was granted funding through this program, which has been matched by U.S. Federal Highway Administration (FHWA) and WSDOT. This Concept of Operations (ConOps) is for work relating to the project objective 1: Improve the binational Cascade Gateway Border Data Warehouse.

What is a Concept of Operations?

A Concept of Operations (ConOps) is a document that provides a high-level identification of user needs and stakeholder agreements on roles and responsibilities for the upgraded border data warehouse. It describes:

- Stakeholder roles and responsibilities
- System needs
- The geographic and physical extent of the system
- Sequence of activities performed
- How the system will be developed, operated, and maintained.
- Needs identified in the ConOps will be used to identify system requirements that will guide design and implementation of the system.

The intended audience for this document includes project stakeholders, funding agencies, consultants working on the project, and end users of the system.

2. Scope

Although the existing warehouse has been running for over four years, the website, the database, and the server have had several maintenance issues which have led to down-time and lost data. Other issues, including an inability to alert administrators to data gaps, slow performance, and difficulty in incorporating new data sets has made improving this warehouse a regional priority.

Upgrades to the warehouse include the following tasks:

Task 1: Transfer existing system to a new platform and upgrade warehouse capabilities

The first task of this project is to upgrade the software used to collect and store data. Years after development of the original archive, the resource suffers from being slow, has maintenance issues, and needs storage capacity and programming improvements for the rapidly increase amounts of data available for the archive. Work under this task will need to resolve the seven identified operational needs (see **Operational Needs**).

Task 2: Develop API interface for other applications to access the warehouse

In addition to gathering, storing, and displaying data from various sources on the website, the new warehouse should also be able to "push" data sets to other interfaces. This task will develop an application programming interface (API) that will allow developers for such tools as Google Gadgets, mobile devices, Facebook, etc. to access the warehouse and display query results or static reports as needed.

The task will not develop the other applications themselves, only the interface to allow other developers to access the data.

Task 3: Document development process

For ongoing maintenance of the warehouse after the initial development, it is critical that excellent documentation of software development, hardware interfaces, communications between agencies, and how to change features exists. A measurement of success in documentation would be the ability for software developers who are familiar with the development platform but who have not worked on the project previously to make changes and improvements to the warehouse without having to learn a new system.

Task 4: Incorporate additional data components to the warehouse

Work under this section is dependent on current efforts of WCOG staff to establish required inter-agency connections and will be offered to the consultant when sufficient details are available to estimate costs and a level of effort. Terms of this work would be negotiated with the consultant separately and should not be included in the proposed cost associated with a response to this RFP.

Since the development of the archive, new data sets that are valuable to regional planners have emerged. The upgraded warehouse needs the capability of storing data sets from different source formats. Currently, WCOG is pursuing access to the following data sets to the warehouse: Commercial Vehicle Information Systems and Networks (CVISN) system data, Weigh-in-Motion (WIM) detector data, and Geographic Positioning System (GPS) truck fleet tracking data. Work will also determine if existing databases such as the U.S. Bureau of Transportation Statistics (BTS) Transborder database may be dynamically shared with the archive to assist end users in finding commodity and travel information. Another possible data set could be the addition of booth status data (open, closed, NEXUS, FAST, auto, truck, etc.) feeds into the system.

Stakeholders

Agencies that will be involved in the project include:

- B.C. Ministry of Transportation (data partnering agency, end user)
- WA State Department of Transportation (data partnering agency, funder, and end user)
- U.S. Federal Highway Administration (funder, end user)
- Transport Canada (funder, end user)
- Whatcom Council of Governments (system administrator, end user)
- University of Washington Advanced Technology Branch (data partnering agency, end user)
- End users (inspection agencies, transportation agencies, tourism organizations, trade organizations, carrier companies, cross-border shippers, customs brokers, transit agencies, researchers, and the traveling public)

3. Referenced Documents

Documentation relevant to this ConOps include:

BC-WA State Cross-Border ATIS Data Management System Project Report *(May, 2007: Whatcom Council of Governments)* – This project report summarized the development of the original warehouse and includes the original work plan and proposal; the XML schema; loop detector translation table and maps; a website user guide; and the project architecture.

BIFA Pilot Project Program Application: Cascade Gateway Border Data Warehouse Upgrade & BIFA Integration *(December, 2009: Whatcom Council of Governments)* – This application for Canadian federal funding details all tasks funded under the BIFA Pilot Project program. This ConOps refers to Objective 1 of the project application.

4. Current System Background

The existing Cascade Gateway Border Data Warehouse stores volume, delay, departure rate, vehicles in queue, and queue length data for passenger vehicles and commercial vehicles northbound and southbound crossing through the four Cascade Gateway land ports-of-entry. 288 packets of data arrive each day (five minute increments of data) for each direction, mode, and crossing. Special lanes like NEXUS and FAST are treated as a separate port in the system for easier analysis.

Data is pulled from two FTP sites, one maintained by IBI Group for BCMOT (which archives a month worth of data on its FTP site for the purposes of this warehouse), and one maintained by WSDOT (which does not archive its data).

Since its inception in January 2007, the archive now holds over 1,300 consecutive days of wait time and volume data , and the types of data being stored continues to expand. At the inception of the project, BCMOT provided auto volume and delay information for southbound Peace Arch and Pacific Highway only (although access to detector data for the truck lanes at Pacific Highway is also stored), and WSDOT provided the same northbound. Since then the system has added northbound auto vehicle data for Lynden/Aldergrove and Sumas/Huntingdon ports-of-entry, northbound Pacific Highway truck wait times and volumes, northbound Pacific Highway FAST lane delay and volumes, and is also storing data for a crossing not in the Cascade Gateway system (State Route 97). In 2011, BCMOT will be expanding its ATIS system to include southbound Lynden/Aldergrove and Sumas/Huntingdon auto volume and delay information as well.

Data from the system is used by multiple regional planning and inspection agencies, including but not limited to U.S. Customs & Border Protection (CBP), Canada Border Services Agency (CBSA), BCMOT, WSDOT, regional chambers of commerce, regional tourism industry representatives, Western Washington University, University of British Columbia, and University of Washington.

Since going live in January, 2007, the warehouse has had over 58,000 unique visitors. Over 300 unique visitors have used the site in excess of fifty times, suggesting regular data queries or ongoing analyses. 66 percent of visitors have come from the BCMOT real-time data site, 16 percent from WSDOT's site, and the rest from a variety of other referrals or search engines.

Most visitors view the data by crossing, although over 7 percent have exported data using the custom query tool. About 2 percent of visitors used the detector viewing component of the warehouse.

In addition to being able to query data on the website www.CascadeGatewayData.com, the warehouse also provides hourly e-mail alerts automatically when a specific delay threshold is reached. For example, an individual may request notification every time southbound delays at Peace Arch or Pacific Highway exceed ninety minutes. An email is sent to them with the hourly delay averages automatically. This resource is currently used by site administrators and border inspection agencies for their own reporting.

Features of the Current System

The existing warehouse offers the following features.

Back-End

- The archive is on a UNIX server and uses Ruby on Rails backed by a PostgreSQL database. The website is presented by Apache.
- A backup drive is separate from the server and backs up all data every twenty-four hours.
- An extract-transform-load (ETL) process is used to update the database.

- An administrative site provides a snapshot view of daily border data file loads, and allows the administrator to update metadata, help sections, establish user accounts, and create e-mail reports.
- E-mail reports can be set up by adding an email address, the ports they are interested in, and the delay threshold.
- Metadata tags data by crossing, direction, and ownership

Front-End

There are three main ways to access the database: by crossing, by detector, or by using the custom query function:

Home Page

- Provides "last updated" information
- Links to partner agency websites, project information, and a usage guide.

By Crossing

- Breaks out each crossing by port, direction, and lane type (NEXUS and FAST are separated)
- Can choose any date or year between 2007 and the present.
- Shows data in a year summary and month summary in calendar format, and day details which include hourly graphs and a data chart.
- Reports can be customized to display departure rate, volume, delay, queue length, and/or vehicles in queue.
- An explanation of measurements and help is available on each page.
- At every data view, data may be downloaded in .CSV format.
- A day-of-week summary is viewable and customizable to show trends by day of week.

By Detector

- Every detector that stores data in the system is individually listed and can be queried to show data in a year, month, and daily summary format.
- Multiple sorting options allow users to find detectors by detector name, station name, lane number, agency, route name, and direction.
- A detector map allows users to pinpoint the exact loop detector on a map and connect to that data set.
- Daily detector reports can be customized to show volume, occupancy, average speed, and average length.

By Custom Query

- Custom queries allow users to download larger amounts of data at once and to specify measurements and multiple locations for customized reports.
- Crossing data and detector data can both be queried by the custom query tool.
- Multiple locations or detectors can be selected at once.

- Output can be grouped by crossing/detector or by direction.
- Any date range can be selected.
- Days of the week can be selected.
- Dates can be grouped by date, day of week, month, or year.
- Time can be grouped by minute, hour, or am/pm.
- Aggregations can be selected for each data field, including sum, average, minimum, maximum, standard deviance, and variance and allow users to show open lanes, service rate (vehicles per five minutes), volume, delay (in minutes), queue length (in meters) and vehicles in queue.
- Data can be exported as a .CSV file or displayed on screen in HTML format.

Limitations of the Current System

Since 2007 the archive has grown in size and the number of crossings and detectors have increased, leading to several issues for the warehouse.

In 2008 the system had a critical breakage but it was difficult to find a developer to repair the database given the platform it was built on. A complete change of the underlying operating system from Windows to UNIX was part of the repair.

Since the warehouse's creation, administrators and users have identified the following limitations:

Automated e-mail system: Few take advantage of the e-mail report system in place because it is not publicly available to site visitors. Automation would allow end-users to subscribe/unsubscribe from e-mail reports themselves, and to specify their own parameters.

Navigation: The site as it is currently designed is not presented in an easy-to-understand manner for the casual end user who is looking for summary statistics of border wait times.

Detector maps: Changes have been made to the loop detector system and new maps need to be generated to show where each loop detector is located. A system should also incorporate real GPS coordinates so that a person can have a more dynamic mapping ability to determine a loop detector location.

Dynamic data views: Dynamic chart manipulation for customized viewing is a feature of many new websites, such as Google Finance tools (i.e. http://www.google.com/finance?q=USDCAD). With the ability to set timelines, change scales, and zoom in on data, these tools allow for more dynamic viewing. The current site limits the viewing to a pre-set scale. More dynamic interfaces would benefit the archive and end user experience.

Section 508 compliance: As per U.S. federal rules, the archive should be fully compliant to section 508 of the Rehabilitation Act of 1973, as amended (29 U.S.C. 794d), which requires electronic and information technology to be accessible to people with disabilities. More information about Section 508 compliance is at: http://www.section508.gov.

Summary statistics: Many users have commented that static reports would be helpful to show one day or one season, comparing "typical" wait times or volumes for that particular period of time. Inspection agencies and tourism bureaus have requested the ability to have summary statistics that automatically update and show average wait times over the course of the day for summer, winter, weekends, week days, holidays, etc. to be usable by the general traveling public.

Printing charts: The current system charts do not print and cannot be used in reports. Being able to share these data is an important feature and should be included in the upgrade.

Integration of NEXUS with port data: NEXUS is a pre-approved traveler program northbound and southbound at the Peace Arch and Pacific Highway ports-of-entry. NEXUS lanes are restricted to travelers enrolled in the NEXUS program (more information about NEXUS at: http://www.getnexus.com). Currently the warehouse treats NEXUS lanes as separate crossings. While this should probably be maintained a method should also be developed to more easily include NEXUS data with the general port data so separate queries don't have to be run on that lane alone.

Southbound FAST data: FAST is the Free and Secure Trade program, a U.S. and Canadian collection of programs designed to expedite the movement of commercial goods across the U.S. – Canada border. FAST lanes operate northbound and southbound at Pacific Highway ports-of-entry and are open only to enrolled FAST drivers, FAST-approved carriers carrying goods from a FAST-approved shipper (more information on the FAST program is available at: http://www.cbp.gov/xp/cgov/trade/cargo_security/ctpat/fast/us_canada). FAST lane data are collected northbound at Pacific Highway. Although detectors are in place southbound at Pacific Highway, no wait times are archived for the FAST system. There may be a way to develop delay figures to store in the archive.

Speed: The greatest challenge currently facing the warehouse is the slow speed of queries given the rapidly increasing size of the database. Over the years the system speed has slowed to the point that it can take several minutes for a web page to load and a long wait for custom queries to run.

Maintenance: The current warehouse was built on a platform that does not lend itself to widespread support and system enhancement options, and it has been difficult to find software developers to assist in maintaining the site.

XML file retention: Currently XML files are generated every five minutes. BCMOT servers store the XML files for at least one month in case the warehouse has issues; however the XML files on the WSDOT server are over-written every day. One possibility to consider is having XML files pushed from WSDOT and BCMOT onto the warehouse server where they could be stored for an additional month or so in case of error.

Volume collection: The system allows administrators to specify the detectors to use as the counters for volume data. However these numbers are often inaccurate compared to monthly volume totals reported by inspection agencies. An improved volume collection system is desired. In addition, if a new port is automatically included in the system, no loop detectors have been set to start collecting volume data and volume is lost until it is manually set up to do so. An alarm would help resolve this issue.

Logs and system updates: Periodic build-ups of system logs, system updates, and other items slow the warehouse down and occasionally break the website.

Missing data: The in-road loop detectors and BCMOT and WSDOT systems as a whole are sensitive to construction and equipment failure. It is difficult with the current system to determine when data gaps occur if the problem is the warehouse, or the lack of incoming data from the supplying agencies. There is also no alerting system to warn when data fails to be downloaded into the archive.

Booth status data: Currently the system does not measure open lanes because this information is not available by detector. However this information would be valuable to end users. In addition, lane status will be a critical component once a variable NEXUS lane is added to the southbound port-of-entry in Peace Arch. The partner agencies will be developing ways to determine lane status, but the system will need to be changed to store these data.

Wait time calculations: Wait times are currently sent to the archive as determined by the northbound and southbound ATIS systems. However if the warehouse itself also had a self-calculated wait time based on arrival rates, open lanes and volume, that number could be stored separately and used for purposes of comparison and validation.

Health check and monitoring: There is no alerting system in place to warn administrators when data fails to be downloaded, or when the site itself is down. A better health-check needs to be in place to identify when a problem occurs and where the problem lies – i.e. is it that the data are unavailable, the data are available and not being understood, or the website is not functioning?

Data quality assessment: The archive does not distinguish between "good," "okay," and "bad" data. Northbound, all data are stored. Southbound, only data tagged "good" are stored. However the system could benefit from a simplified color-coded system for each day or month on the day/month view to determine the number of data packets calculations are based on, and the quality of those data packets.

File transfers: BCMOT and WSDOT have both changed their data output formats, data storage locations, and other details since the project inception in 2007 and it has been difficult to make rapid system alterations to respond to these changes. In addition, BCMOT recently removed FTP and instead switched to HTTP transfers, which has required patches for the existing system. A better way to interface with the BCMOT and WSDOT servers, and a better way to update the way these connections work, needs to be developed.

Adding new ports difficult: Although the XML schema is designed to automatically categorize and store data from new ports-of-entry, the administrator is not alerted to the presence of new data, and some port data has had to be customized for the system to read it. Port data does not automatically show up on the system and must be programmed to show by the developer.

Growing storage requirements: The archive is growing every day. Although there is plenty of storage capacity on the server, the functionality of the archive database is affected as the amount of data to be queried grows, making the system inefficient.

Justification for Changes

Given the widespread use of the website by multiple agencies for historic wait time and volume data, the priority of the warehouse by regional agencies and funding partners, and the investment already made into the system, it makes sense to update the system to address the limitations of the current warehouse. Given that incoming data is increasing and there are even more opportunities for better data collection and dissemination, it is critical to keep this archive healthy and functioning.

5. Concept for the Proposed System

Concepts examined to address the current limitations of the warehouse include:

- Find a developer to work with the existing warehouse platform.
- Find a developer to create a new warehouse based on the existing warehouse and preserving the original data.
- Not changing the existing system.

The first option would allow the system to function as is and minimize the need to involve stakeholders (WSDOT and BCMOT particularly) as it builds on the existing design. However this does not resolve the issue of finding a variety of developers for maintaining the system.

The second option requires more work at the beginning and the full participation of project stakeholders. BCMOT and WSDOT, as well as stakeholders representing the new data sets (i.e. WIM managers and GPS fleet data representatives) would be engaged throughout the project to design it to meet their requirements. Although this takes more work up-front it will lead to a more stable warehouse that should address all of the concerns listed in the needs section of this ConOps.

The third option is to not change the system, which currently functions but not optimally. Given the increasing importance of the warehouse to a growing number of agencies and individuals, this is not a viable alternative.

The concept for the proposed system, therefore, is to redevelop the warehouse from scratch, although using the existing warehouse's archived data and design as a template to build upon.

Steps would include:

- Review and select a system platform that will be easier to maintain and build upon over the next ten years.
- Select system developers that will redesign the archive to optimize speed, storage, and web access, with a focus on documenting all changes, simplifying updates, automatic new data flows, and creating an API interface to allow the data to be used in multiple formats.
- Work with stakeholders to streamline the data sharing process and how information is to be loaded into the archive.

6. User Oriented Operational Description

A description of the roles and responsibilities of stakeholders in the current system can be found in the **existing project architecture** (see related documentation). A summary is below:

WCOG, as the lead agency for IMTC and the project manager of the original warehouse development, maintains the website; maintains the archive; owns the server; and is responsible for working with the other stakeholders to assure on-going functionality of the warehouse.

BCMOT provides data from its ATIS system through its consultant, IBI Group. They are responsible for uploading the data to the FTP location daily. They are also responsible for maintaining the in-road hardware used to collect the data.

WSDOT provides data from its ATIS system by uploading data to the FTP location daily. They are also responsible for maintaining the in-road hardware used to collect the data.

Data sharing operations and how the agencies interact is all included in the project architecture.

7. Operational Needs

Given the limitations of the current system and discussions with stakeholders and system users, the following needs have been identified to improve the current warehouse:

1. The warehouse needs to maintain existing features (including but not limited to port and detector data, e-mail automated reports, downloadable files, and custom queries).

2. The warehouse needs to present data in a clear, concise manner that is easy for inexperienced end-users to find what they are looking for.
3. The warehouse needs to provide data in downloadable and exchangeable formats that are easy to understand and capable of being used on multiple platforms.
4. The warehouse needs to quickly and efficiently pull data from the archive.
5. The warehouse (website and data archive) needs to be developed in such a way as to make it simple to modify and maintain in the future.
6. The warehouse needs to be designed in a way that administrators can monitor the health of the warehouse and be alerted of errors.
7. The warehouse needs to be built for future expansion and increased archiving requirements.

Note that these operational needs all trace to the original limitations list of the current archive, and to the list of requirements for the development of the upgraded site (**see Traceability Matrix**).

8. System Environment

Geographic Environment

The existing warehouse currently collects data from the Cascade Gateway ports-of-entry. It also stores data from State Route 97, which has wait time and volume information from WSDOT's system that is not being stored anywhere else at present.

The expanded warehouse will change the geographic scope slightly to include the approach roads to the border crossings. Specifically, data from the Interstate 5 corridor and from B.C. Highway 15 will be included.

Operational Environment

The physical server of the warehouse is located in Bellingham, Washington at a private internet service provider (ISP). WCOG is a client of the ISP and houses the server in their protected facility. Although the ISP maintains the server, the server itself is owned by WCOG.

The warehouse is maintained over an internet connection by WCOG in Bellingham, Washington.

The hardware used to collect the data is maintained by the participating agencies. Specifically, BCMOT and WSDOT own their own hardware and equipment.

Support Environments

The current warehouse is maintained by personnel at WCOG; a software developer in Bellingham, Washington who is used on an as-needed basis for repairs; and the ISP company in Bellingham, Washington who maintain the physical server box.

9. Operational Scenarios

Operational scenarios convey the concept of the upgraded warehouse to help audiences understand what is expected to be achieved with the implementation of the upgraded system. The following scenario have been developed with project stakeholders to illustrate how the warehouse will function.

Typical Use Scenario

A researcher is looking to collect average delay at several border crossings to feed this data into a transportation model. At www.CascadeGatewayData.com he navigates to the custom query tool.

Using the custom query tool, he selects which ports-of-entry and directions he is interested in, and how he would like the data to be grouped. He selects the date range, day of week, and date and time groupings he is interested in. He then selects how data will be aggregated, choosing to see average delay only. He exports the data in .CSV format and is able to view it and use it in a spreadsheet format.

New Static Report Scenario

A cross-border traveler is trying to determine the best time of day to cross the border on a summer weekend. She visits the website www.CascadeGatewayData.com, or an affiliated partner website that has a live feed of the border warehouse static reports.

At the website, she navigates to the section providing typical delay information and trends, and selects "summer weekend" as her travel period.

Charts and graphs are provided that show travel trends for summer weekend travel, including average delay and average volumes, for each direction, and each port. Using the report, she can determine the best crossing to use for the time of day she plans to cross the border. If she wants, she can also print or download the report as a .CSV file or as a PDF.

New Email Report Scenario

An individual working on cross-border inspection or transportation issues wants to be alerted every time delays at the Peace Arch crossing exceed one hour.

He goes to the website www.CascadeGatewayData.com and navigates to the "sign up for e-mail reports" section of the site. He enters in his e-mail address and password if he has already registered for reports; if not, he sets up a new account, entering in his name, agency, e-mail address, and selecting a password.

He then can select which crossings he is interested in reports for, which directions, and what the delay threshold he would like to set (chosen by minutes). He can also choose to have reports when volumes reach a set threshold. He then selects whether he wants a report for every hour the threshold is met, or an omnibus report sent once every twenty-four hours.

Once set up, the system will send an e-mail to his address that includes statistics by hour for each of the crossings and directions he has selected.

If he decides he no longer wants to receive this report, or wants to change the reporting, he can log back on to the system and edit the reports he currently has selected.

10. Summary of Impacts

Because the upgraded warehouse will be building on existing relationships with stakeholder agencies, the impacts to partner agencies should remain minimal.

WCOG: WCOG will continue to maintain the website and serve as the warehouse administrator. The upgraded system should make maintenance easier, and expedite efforts to address missing data thanks to missing data alarms in the new system. There may be additional work required with the increased number of data-sharing partners in the upgraded warehouse.

BCMOT: Current FTP protocols and data storage should remain unchanged; the addition of new systems should be seamless. However, the addition of previously un-stored WIM data will require adding communication and powering devices to the WIM unit as well as the conversion of the data from its original format into a format understandable by the system as per the established XML schema.

WSDOT: Similarly, the only new component to the upgraded system will be the addition of WIM detector data. Hardware may need to be added to power and add communication to the unit, and data formats may need to be converted to fit the schema.

CVISN: The CVISN detectors will need to be powered and have their data converted to apply to the XML schema. The impacts this may have are currently unknown.

GPS Data: Although a new stakeholder, this is a purchased data set that may or may not be automated. The impacts this may have are currently unknown.

BTS: The Bureau of Transportation System database managers will have to work with project partners to identify whether automated data downloads can be achieved and if changes are needed in the BTS database.

Evaluation Metrics

The following are potential measures of the upgraded system's performance:

- Overall speed of the database
- Presentation of the data in clear and concise format
- Number of individuals signed up for automated e-mail alerts
- Functionality of administrative back-end
- Easy of updating and maintaining warehouse
- Data download consistency and accurate logging

11. List of Abbreviations

ATIS	Advanced Traveler Information Systems
BCMOT	B.C. Ministry of Transportation
BIFA	Border Information Flow Architecture
BTS	U.S. Bureau of Transportation Statistics
CBSA	Canada Border Services Agency
CBP	U.S. Customs & Border Protection
CVISN	Commercial Vehicle Information Systems and Networks
ETL	Extract-transform-load
FHWA	U.S. Federal Highway Administration

FTP	File Transfer Protocol
GPS	Geographic Positioning System
IMTC	International Mobility & Trade Corridor Project
ISP	Internet service provider
ITS	Intelligent Transportation Systems
MPO	Metropolitan Planning Organization
RFP	Request for Proposals
SR	State Route
WCOG	Whatcom Council of Governments
WIM	Weigh-in-Motion
WSDOT	WA State Department of Transportation

12. List of Current and Planned Data Sets

The following tables compare existing and planned data sets for this project:

EXISTING DATA SETS						
Source	**Crossing**	**Route**	**Direction**	**Vehicles**	**System**	**Data Type**
WSDOT	Peace Arch	I-5	NB	Cars	Loops	Volume, delay, queue length, departure rate
WSDOT	Peace Arch	I-5	NB	NEXUS Cars	Loops	
WSDOT	Pacific Highway	SR543	NB	Cars	Loops	
WSDOT	Pacific Highway	SR543	NB	NEXUS Cars	Loops	
WSDOT	Pacific Highway	SR543	NB	Trucks	Loops	
WSDOT	Pacific Highway	SR543	NB	FAST Trucks	Loops	
WSDOT	Lynden/Aldergrove	SR539	NB	Cars	Loops & LPRS	
WSDOT	Sumas/Huntingdon	SR9	NB	Cars	Loops & LPRS	
BCMOT	Peace Arch	HWY99	SB	Cars	Loops	
BCMOT	Peace Arch	HWY99	SB	NEXUS Cars	Loops	
BCMOT	Pacific Highway	HWY15	SB	Cars	Loops	
BCMOT	Pacific Highway	HWY15	SB	NEXUS Cars	Loops	

PLANNED DATA SETS						
Source	**Crossing**	**Route**	**Dir.**	**Vehicles**	**System**	**Data Type**
WSDOT	Lynden/Aldergrove	SR539	NB	Trucks	Loops	Volume, delay, queue length, departure rate
WSDOT	Sumas/Huntingdon	SR9	NB	Trucks	Loops	
WSDOT	Pacific Highway	I-5	NB	Trucks & Cars	WIM	Volume, classification, empty/loaded

PLANNED DATA SETS						
Source	**Crossing**	**Route**	**Dir.**	**Vehicles**	**System**	**Data Type**
WSDOT	Pacific Highway	I-5	SB	Trucks & Cars	WIM	
WSDOT	I-5 Corridor	I-5	NB	Trucks	Transponders	Location, travel time, weight
WSDOT	I-5 Corridor	I-5	SB	Trucks	Transponders	
BCMOT	Pacific Highway	HWY15	SB	Trucks	Loops	Volume, delay, queue length, departure rate
BCMOT	Pacific Highway	HWY15	SB	FAST Trucks	Loops	
BCMOT	Lynden/Aldergrove	HWY13	SB	Cars	Loops	
BCMOT	Lynden/Aldergrove	HWY13	SB	Trucks	Loops	
BCMOT	Sumas/Huntingdon	HWY11	SB	Cars	Loops	
BCMOT	Sumas/Huntingdon	HWY11	SB	Trucks	Loops	
BCMOT	Pacific Highway	HWY15	SB	Trucks & Cars	WIM	Volume, classification, empty/loaded
BCMOT	Pacific Highway	HWY15	NB	Trucks & Cars	WIM	
BCMOT	Lynden/Aldergrove	HWY13	SB	Trucks & Cars	WIM	
BCMOT	Lynden/Aldergrove	HWY13	NB	Trucks & Cars	WIM	
BCMOT	Sumas/Huntingdon	HWY11	SB	Trucks & Cars	WIM	
BCMOT	Sumas/Huntingdon	HWY11	NB	Trucks & Cars	WIM	
GPS	Pacific Highway	HWY15	NB	Truck	GPS	Location, travel time
GPS	Pacific Highway	HWY15	SB	Truck	GPS	
GPS	Pacific Highway	I-5	NB	Truck	GPS	
GPS	Pacific Highway	I-5	SB	Truck	GPS	
GPS	Lynden/Aldergrove	HWY13	NB	Truck	GPS	
GPS	Lynden/Aldergrove	HWY13	SB	Truck	GPS	
GPS	Lynden/Aldergrove	SR539	NB	Truck	GPS	
GPS	Lynden/Aldergrove	SR539	SB	Truck	GPS	
GPS	Sumas/Huntingdon	HWY11	NB	Truck	GPS	
GPS	Sumas/Huntingdon	HWY11	SB	Truck	GPS	
GPS	Sumas/Huntingdon	SR9	NB	Truck	GPS	
GPS	Sumas/Huntingdon	SR9	SB	Truck	GPS	
BTS	Pacific Highway		NB	Freight	BTS Database	Freight by value, commodity
BTS	Lynden/Aldergrove		NB	Freight	BTS Database	
BTS	Sumas/Huntingdon		NB	Freight	BTS Database	
BTS	Pacific Highway		SB	Freight	BTS Database	Freight by value & weight, by commodity
BTS	Lynden/Aldergrove		SB	Freight	BTS Database	
BTS	Sumas/Huntingdon		SB	Freight	BTS Database	

Appendix B:
Summary of Feedback for the Cascade Gateway Border Data Warehouse

1. Background

The information and comments that are included in this summary were gathered from an online survey distributed to approximately fifty agencies considered stakeholders for the Cascade Gateway border crossings, and from a Project Advisory Team meeting.

The July 6, 2010 Advisory Team Meeting

The July 6 project meeting included representatives from BC. Ministry of Transportation, WA State Department of Transportation, Transport Canada, and the WA State Transportation Research Center. Participants were asked to discuss limitations to the existing border data warehouse, and what features they would like to see improved and added.

The Online Survey

Using *Surveymonkey.com*, WCOG solicited and compiled assessments of the existing warehouse and suggestions for improvements and additions to the upgraded warehouse.

There were 32 respondents to the survey of whom 47 percent (15 respondents) regularly use the existing warehouse. The summarized feedback is based on the fifteen questionnaires.

2. Evaluation of the Current Warehouse

Survey participants responded to questions about how frequently they visit the site, what data they access, and how they use the information.

Frequency of use — Of the current users who responded to the survey most reported that they only use the site a few times a year or have only used the site a few times. A user commented that the site was too slow to be easily accessed, although another participant though that the site would be a "valuable resource" for an upcoming project.

Reasons for using the warehouse --- Most users visited the warehouse for wait time information and traffic volume data. This information was used to evaluate traffic staging strategies, determine border wait times during the 2010 Vancouver Olympic Games, for presentations, and for policy development.

Ports of interest — Use of the warehouse is spread evenly over all of the available ports-of-entry (Peace Arch northbound/southbound passenger vehicles and NEXUS passenger vehicles; Pacific Highway northbound/southbound passenger vehicles, NEXUS passenger vehicles, and trucks; Lynden-Aldergrove northbound passenger vehicles; and Sumas-Huntingdon northbound passenger vehicles). Users requested more information about southbound trucks at Pacific Highway and for integrated NEXUS lane data.

Usefulness of .CSV download format — Few participants said that they had used the system's ability to export comma-separated-value (.CSV) data, although they felt it was an important and valuable option.

Custom query tool — Users were satisfied or mostly satisfied with the information they were able to access with the custom query tool. Users reported querying several days' worth of data or data about a specific port and time. Some respondents didn't like the slow speed and missing data in the query results.

Data detectors — All participants who had used individual detector data had accessed the information via the detector maps. Most comments requested that actual images or scaled drawings replace the current detector maps.

3. Recommended Changes and Improvements

Recommendations of survey respondents and the advisory team are grouped below by system functionality, data improvements or data additions.

Functionality

Improved speed —19 percent of respondents cited "improved speed" as their top priority for the new warehouse. Some also noted that the site's slowness diminished its usability.

Data quality flags — Respondents suggested adding a rating system of the archived data so that users know the completeness and quality of the information. The flags would be based on the number of data packets received per month or day and on the quality of those data packets.

Health check and monitoring system — It was recommended that the new system have a better set of alarms to alert administrators to error and to identify the cause of missing data.

Volume data – While volume data were listed as an important feature of the site discussions held with the advisory team noted that the volume data itself may need to be validated. Volumes often differ across inspection agency sources. In a related project, the Whatcom Council of Governments will be reviewing the sources and definitions used in volume data from all agencies who collect cross-border traffic information. This information will help identify needed corrections and opportunities for improved accuracy.

New server and processor — To help speed the current system, it was recommended that the project purchase a top-of-the-line server with fast processors and capacity for growing storage needs.

Have XML files stored longer — Currently XML files are generated every five minutes. The archive uses FTP to fetch the files from the WSDOT and MOT servers. MOT servers store the XML files for at least one month in case the warehouse has issues; however the XML files on the WSDOT server are over-written every day. One suggestion was to have the XML files pushed from WSDOT and MOT onto the warehouse server and be stored there for an additional month or so in case of error.

Consider alternatives to FTP — Since the design of the original archive, alternatives to using FTP to automate data exchanges between servers have been developed. It was recommended that the software developers hired for the project look at other technologies available for exchanging the data.

Print feature for charts — The current system charts do not print and cannot be used in reports. Being able to share these data is an important feature and should be included in the upgrade.

Improved Data Sets

The advisory team noted a need for two types of output for two audiences: more detailed, improved data sets for specific queries to be used by power-users; and less detailed, more summarized static data reports that provide quick answers to common questions.

Static reports —Would show trends but not much detail and allow for integration with other regional research. Year-by-year comparisons, peak travel times, monthly and yearly volume trends were some of the suggested static reports.

More dynamic data views — Respondents requested dynamic chart manipulation for customized viewing to be able to set timelines, change scales, and zoom in on data.

Better integration of NEXUS with port data — Currently NEXUS lanes are treated as separate crossings in the ATIS systems. An easier way to include NEXUS volume data in the general port data was requested so separate queries don't have to be run on that lane alone.

BTS data link — A request for an easy link to regional cross-border commodity data from the U.S. Bureau of Transportation Statistics (BTS) was noted. There is interest in having BTS data automatically feed into the archive for easier use. If a direct interface with BTS isn't possible, another suggestion was to add a "my favorite" query link, so that the system saves parameters set in the BTS query form and populates that form automatically for users. This link would allow users to more easily find commodity and trade value data.

Active lanes data — Both the advisory team and survey respondents requested more information about the number of open lane booths at each crossing. This additional information would enhance models and GIS applications, and to analyze cross-border traffic flow.

New Data Sets

In addition to suggesting improvements to existing data, respondents made requests for additional data to be included in the upgraded warehouse:

Archive-calculated wait times — Wait times are currently sent to the archive as calculated by the northbound and southbound systems respectively. A suggestion was made to have the archive database calculate its own wait times for comparison and to have the same formula applied to both northbound and southbound traffic flow. This would be a separate value from the wait times estimated and archived from the state and provincial systems.

Comparison graphs from the custom query — Year by year comparisons, peak travel times, monthly and annual volume trends were some of the suggested outputs.

New maps needed — Many users agreed that new maps of the detectors and other equipment in the ATIS systems should be developed, including better location orientation and scaled images. Also, clicking on a detector should pull up that detector's data in the new warehouse site.

Automated e-mail reports — Although the current system does have an e-mail delay report feature, because it has to be set up by the site administrators and individuals cannot subscribe or unsubscribe to the service, it is not used by many. It was recommended that this feature be automated for users and that it also be part of a marketing effort to publicize the archive and promote this feature. Additionally, stakeholders asked to be able to customize which reports or updates they receive by e-mail. Most of those surveyed (80%) said that they would be interested in these e-mailed reports.

Greening the Border system data — The traffic lights southbound at Peace Arch are tied into the existing ATIS systems and facilitate the operation of an anti-idling zone for queued cars at the border. The system generates traffic signal cycle times, which are determined by the queue length in front of the booth. This cycle length data could be stored by the new system.

FAST Lane data — FAST Lane data are collected northbound at Pacific Highway but not southbound, although loop detectors are in the roadway and could be collecting the wait times. However, given the potential changes for FAST lane use at the Pacific Highway crossing, it may not be worth investing effort to develop these delay figures for the archive.

Truck wait times — Currently the ATIS systems only calculate truck wait times northbound at Pacific Highway. Although the southbound expansion plans for the BCMOT ATIS system do not include truck wait times, loop detectors will be installed in truck lanes as they are at Pacific Highway. The northbound truck wait times are based simply on queue and service rate, and do not include trucks that wait in the parking area. Wait time data for trucks could be archived, even if it isn't displayed on signage.

Predictive calculations — A question was raised about the interest in developing predictive wait times. At this point there is enough historic data to develop an initial effort. In the initial feedback for the project there was interest from inspection agencies in estimating future wait times. It was suggested that this be pursued as a separate project and that the system be tested for approximately a year before rolling out anything publicly.

General System Expandability — The system should be able to expand to include additional data sets, including predictive data, truck wait times and car wait times at other crossings, and possibly the inclusion of bus data.

4. Summarized Online Survey Responses

The following charts illustrate summarized responses from the online survey. More details are available through the Whatcom Council of Governments.

1. How often to you use the border data warehouse at CascadeGatewayData.com?

At least once a month	3.1%
A few times a year	25.0%
Only a few times	18.8%
Never	53.1%

2. Do you ever have a need for cross-border traffic volume or freight data?

Yes	52.9%
No	47.1%

3. Please provide a description of what kind of cross-border traffic data you are interested in:

- Commercial volumes and wait times
- Value of cross-border cargo
- Freight and freight vs. vehicle origin & destination data

4. What were your reasons for visiting the warehouse? (Check all that apply)

Wait time (border delay) data	80.0%
Traffic volume data	66.7%
Arrival/departure rate data	26.7%
Other	6.7%

5. What ports-of-entry are you most interested in data for? (Check all that apply)

Peace Arch southbound	60.0%
Peace Arch northbound	60.0%
Peace Arch NEXUS lanes	60.0%
Pacific Highway southbound	66.7%
Pacific Highway northbound	73.3%
Pacific Highway NEXUS lanes	46.7%
Lynden/Aldergrove northbound	46.7%
Sumas/Huntingdon northbound	53.3%

6. Please provide an example of how you have used data from the border data warehouse:

- As a Customs broker we use the data to compare to our records of customs clearances
- Evaluate traffic staging strategies
- Tourism marketing
- During the Olympics we used wait times from the border data warehouse rather than our own times because the data we send is screened to the minimum of 5 minutes, and our data just saves the raw number which could often be zero
- Looking for the best time to cross with least delays
- We've summarized the data and incorporated findings into presentation materials
- In communicating border issues to others in and out of the county/state
- We use any and all time-series data related to trucks for presentations and to develop policy. We also refer media, researchers, government officials, and others seeking data to the data housed on your website
- Determining service times at booths
- Use to populate GIS website for a DHS study

7. Have you ever used the .CSV format to download data from the website?

Yes	53.3%
No	46.7%

8. Have you ever used the custom query tool on the website?

Yes	53.3%
No	46.7%

9. How satisfied were you with the custom query tool?

Very satisfied	11.1%
Satisfied	33.3%
It was all right	44.4%
Not satisfied	11.1%

10. Can you give an example of a custom query you have run using the custom query tool? (optional)

- S/B Hwy 99 daily and weekly volume. Delay and queue length.

11. Do you ever look for data on the warehouse by detector?

Yes	46.7%
No	53.3%

12. Please explain how you use the data from the detectors:

- Checking system health and performance for BCMoT S/B ATIS
- For queue analysis
- To put in GIS package

13. How did you find the detector(s) you were looking for?

By the loop detector maps	100%
By the list of detector names	0%

14. Please describe how the maps worked for you and what changes you would like made:

- Locational referencing
- Maybe we could overlay detector location on Google maps for easier orientation
- Seeing accurate loop locations on aerial photo with scale
- Maps worked well

15. Do you subscribe to automatic email reports when delays reach a certain threshold?

Yes	0%
No	100%

16. How satisfied are you with these reports?

Very satisfied	0%
Satisfied	0%
They were all right	0%
Not satisfied	0%

17. Would you be interested in a feature that would allow you to subscribe and unsubscribe on the website to email reports? You would be able to set the parameters of when you would receive emails by port and wait time (i.e. send a report every time waits at Peace Arch exceed 60 minutes).

Yes	40.0%
Maybe	40.0%
No	20.0%

18. What improvements or additional features would you like to see on the archive? (Please rank in order of importance, with 1 being most important):

	1 (First)	2	3	4	5	6 (Last)
Improved speed	**38.5%**	7.7%	7.7%	23.1%	15.4%	7.7%
More detail about the data	8.3%	16.7%	**33.3%**	25.0%	8.3%	8.3%
More static reports	7.7%	**38.5%**	30.8%	7.7%	15.4%	0.0%
Improved mapping	**23.1%**	**23.1%**	0.0%	**23.1%**	**23.1%**	7.7%
Additional data sets	**30.0%**	10.0%	20.0%	20.0%	10.0%	10.0%
Automated email reporting	7.7%	7.7%	23.1%	0.0%	7.7%	**53.8%**

19. Other comments about the displays or functionality of the website:

- More static and more customs choices
- It would be nice to generate comparison graphs from a custom query
- The automated emails should contain the options for when data is lost.

20. What additional data would you like to see made available on the archive?

- Number of inspection lanes open
- Better truck data

21. Please describe what static reports you would like to see made available? (i.e. monthly wait time averages by border crossing and weekday/weekend day)

- Number of hours when wait time is over __ minutes, and the distribution of these periods of time with lengthy wait time
- Monthly and yearly volumes by crossing and direction.
- Monthly and yearly total volumes (all four crossings) by direction

- Best and worst times to cross
- Year-by-year comparisons

22. Would you be interested in a feed of this data to other applications (i.e. phone apps or widgets) or to other websites?

Yes	42.9%
No	57.1%

Appendix C:
System Requirements Traceability Matrix for the Cascade Gateway Border Data Warehouse

1. Introduction

The Cascade Gateway Border Data Warehouse is a U.S. – Canadian data storage server and corresponding database that archives real-time traffic data relating to the Cascade Gateway system of border crossings.

To define the work needed to address system limitations and improve the warehouse, a feedback survey and stakeholder interviews defined seven basic operational needs to address limitations in the current system:

1. The warehouse needs to maintain existing features (including but not limited to port and detector data, email automated reports, downloadable files, and custom queries).
2. The warehouse needs to present data in a clear, concise manner that is easy for inexperienced end-users to find what they are looking for.
3. The warehouse needs to provide data in downloadable and exchangeable formats that are easy to understand and capable of being used on multiple platforms.
4. The warehouse needs to quickly and efficiently pull data from the archive.
5. The warehouse (website and data archive) needs to be developed in such a way as to make it simple to modify and be alerted of errors.
6. The warehouse needs to be designed in a way that administrators can monitor the health of the warehouse and be alerted of errors.
7. The warehouse needs to be built for future expansion and increased archiving requirements.

2. Developing the matrix

Using these operational needs as the starting point, both high-level and detailed system requirements were defined in a matrix that would allow project managers to trace each system improvement back to an original operational need.

In addition, the matrix was developed to serve as the validation documentation for systems engineering.

The matrix was approved by the project management and used by the consulting team developing the archive and website.

<table>
<tr><th colspan="2">DETAILED REQUIREMENTS</th><th colspan="2">HIGH-LEVEL REQUIREMENTS</th><th colspan="2">OPERATIONAL NEEDS</th><th>LIMITATIONS IN CURRENT SYSTEM</th></tr>
<tr><td>1.1.1</td><td>The data shall be presented in calendar format.</td><td rowspan="5">1.1</td><td rowspan="5">The warehouse needs to present data in a similar fashion to the existing website.</td><td rowspan="8">1</td><td rowspan="8">The warehouse needs to maintain existing features (including but not limited to port and detector data, e-mail automated reports, downloadable files, and custom queries).</td><td rowspan="8">Non-automated email system</td></tr>
<tr><td>1.1.2</td><td>The data shall be viewable by port-of-entry.</td></tr>
<tr><td>1.1.3</td><td>The data shall be viewable by direction.</td></tr>
<tr><td>1.1.4</td><td>The data shall be viewable by either detector or crossing.</td></tr>
<tr><td>1.1.5</td><td>The data shall be queriable by custom query.</td></tr>
<tr><td>1.2.1</td><td>The website shall allow users to sign up for automated email reports.</td><td rowspan="3">1.2</td><td rowspan="3">The warehouse needs an automated sign-up for email reporting that users can sign up for and define the thresholds for reporting themselves.</td></tr>
<tr><td>1.2.2</td><td>The website shall allow users to define the type of reports they are interested from a set of options.</td></tr>
<tr><td>1.2.3</td><td>The website shall allow users to unsubscribe from reports.</td></tr>
<tr><td>2.1.1</td><td></td><td rowspan="2">2.1</td><td rowspan="2">The website needs a better user guide online and help section.</td><td rowspan="12">2</td><td rowspan="12">The warehouse needs to present data in a clear, concise manner that is easy for inexperienced end-users to find what they are looking for.</td><td rowspan="12">Difficult for casual end users / loop detector maps dated/ lacks dynamic charts/ not fully Section 508 compliant</td></tr>
<tr><td>2.1.2</td><td>The website administration shall allow administrators to write and edit help sections.</td></tr>
<tr><td>2.2.1</td><td>The website shall separate out data sets using tabs at the top of the page for each data type.</td><td rowspan="3">2.2</td><td rowspan="3">The warehouse needs to incorporate the new data sets in a clear manner.</td></tr>
<tr><td>2.2.2</td><td>The website shall develop the visual display of each data type in a similar manner.</td></tr>
<tr><td>2.2.3</td><td>The website shall be designed in such a way that a finite number of data sets can be displayed at the top of the page.</td></tr>
<tr><td>2.3.1</td><td>The website shall use an online mapping system to display detectors.</td><td rowspan="5">2.3</td><td rowspan="5">The warehouse needs to have geo-coded detector and system maps for users to more easily access port-level data and individual detector data.</td></tr>
<tr><td>2.3.2</td><td>The website shall use an online mapping system to display port-of-entry crossing symbols.</td></tr>
<tr><td>2.3.3</td><td>The website mapping shall be zoomable and allow end users to pick specific detectors from the map itself.</td></tr>
<tr><td>2.3.4</td><td>The website shall allow administrators to input the geo-location of each detector and automatically display this detector on the map.</td></tr>
<tr><td>2.3.5</td><td>The website shall allow administrators to edit the locations of detectors and crossings using an editing</td></tr>
</table>

DETAILED REQUIREMENTS		HIGH-LEVEL REQUIREMENTS		OPERATIONAL NEEDS		LIMITATIONS IN CURRENT SYSTEM
	tool.					
2.4.1	Website data shall be displayed in data tables and in charts.	2.4	The website chart views needs to be dynamic and allow for end-user customization of scales, timelines, and comparisons.			
2.4.2	Website charts shall be designed in a way to allow users to customize the visual display.					
2.4.3	The website chart views shall be printable.					
2.5.1	The website shall be designed in such a way that NEXUS port data and similar direction/port data can be displayed separately or together.	2.5	The website data views need to allow for easy combinations of port data (i.e. NEXUS lane data with the rest of the port data) for simpler queries.			
2.6.1	The website needs to be developed as per section 508 of the Rehabilitation Act of 1973 for accessibility for all users.	2.6	The website needs to be developed as per section 508 of the Rehabilitation Act of 1973, as amended (29 U.S.C. 794d) for accessibility for all users.			
2.6.2	The website developers shall instruct the administrators on how to maintain the website in such a way as to continue to keep all aspects of the site accessible to all users.					
3.1.1	The website shall have a tab for reports.	3.1	The website should include static reports which summarize typical wait times and how a particular day compares to an average of such days. This requirement can be handled by way of "Saved Queries" for which the results can be printable.	3	The warehouse needs to provide data in downloadable and exchangeable formats that are easy to understand and capable of being used on multiple platforms.	No summary statistics/charts don't print/ No FAST/NEXUS integration
3.1.2	The reports shall be configurable and organizable by website administrators.					
3.1.3	The reports shall be viewable on screen or downloadable as .csv or equivalent file types.					
3.1.4	The website shall be able to list any number of canned reports for viewing.					
3.1.5	The reports shall be designed in such a way that they are always kept current (i.e. new data is included in the report query).					
3.2.1	The website shall include a custom query tool for end users to run unique reports on all sets of data.	3.2	The custom query tool needs to allow for different aggregations, including a daily time output (in addition			
3.2.2	The custom query shall result in multiple output types including data table, chart, and downloadable file.					

DETAILED REQUIREMENTS		HIGH-LEVEL REQUIREMENTS		OPERATIONAL NEEDS		LIMITATIONS IN CURRENT SYSTEM
3.2.3	The custom query tool shall allow users to select date ranges and days of the week.		to minute, hour, and am/pm outputs).			
3.2.4	The custom query tool shall allow users to group by month, day of week, day, am/pm, hourly, or by five minute increment.					
3.2.5	The custom query tool shall allow users to query multiple detectors or crossings at a time.					
3.2.6	The custom query tool shall allow users to query by holiday.					
3.2.7	The custom query tool shall allow users to aggregate data by various measures, including average, max, min, and sum values.					
3.2.8	The custom query tool shall allow users to view multiple measures of data in one query (i.e. volume, wait times, etc).					
3.3.1	The warehouse shall have an associated interface to allow data to be shared with other websites.	3.3	The warehouse needs an API interface to allow the data to be "pushed" to other websites, agencies, or applications for greater dissemination of data.			
3.3.2	The warehouse API shall be designed in such a way as to maintain the security and integrity of the original database.					
3.4.1	The website shall produce viewable charts that can be printed; and the associated data can be downloaded in CSV format	3.4	The website charts need to be able to be printed and downloaded for end-user reports.			
3.5.1	The warehouse data shall be available in downloadable formats including but not limited to .csv format.	3.5	The warehouse data needs to be able to be downloadable in .csv format.			
4.1.1	There shall be no more than a five second delay for each page load on the website.	4.1	The warehouse access speeds need to be improved.	4	The warehouse needs to quickly and efficiently pull data from the archive.	Speed (lack thereof)
4.1.2	Large custom queries shall take no longer than ten seconds to load and display.					
4.1.3	The website shall provide a loading or waiting symbol for any query or page load that takes longer than a few seconds.					

DETAILED REQUIREMENTS		HIGH-LEVEL REQUIREMENTS		OPERATIONAL NEEDS		LIMITATIONS IN CURRENT SYSTEM
4.2.1	The warehouse shall be stored on a server with enough capacity to house increasing volumes of data.	4.2	New hardware needs to be identified to facilitate speedy data queries and increasing storage requirements.			
4.2.2	The warehouse shall be stored on a server designed in such a way that website function is separated from data storage to facilitate faster response times.					
4.2.3	The warehouse data backups shall be stored in a location that will not impact the speed or functionality of the website.					
5.1.1	The website developers shall develop documentation on how the warehouse has been developed; hardware specifications; programming knowledge needed; and how and where data are stored.	5.1	Documentation needs to be developed so the functioning of the warehouse is clear and specific for future developers.	5	The warehouse (website and data archive) needs to be developed in such a way as to make it simple to modify and maintain in the future.	Maintenance (difficult)/ files deleted
5.1.2	The website developers shall prepare documentation on how the warehouse should be maintained.					
5.1.3	The website developers shall prepare documentation on how administrators can make changes to the website.					
5.2.1	The website shall be designed using a common programming language and on a common server operating system.	5.2	Database and website formats need to be in an accessible programming language that allows for multiple developers to easily understand and address future issues.			
5.2.2	Any customized components of the warehouse software shall be described in detail in documentation with instructions on how that component ties into the full warehouse/website.					
5.3.1	The warehouse administration shall allow administrators to add or edit data feeds into the system.	5.3	System needs an easy way to incorporate changes to data transfer technologies or locations so that the administrator can manually change data interfaces.			
5.3.2	The warehouse administration shall allow administrators to be alerted if changes occur to the data feeds in the system.					
5.4.1	The warehouse shall preserve XML files downloaded from partner agencies in a backup location no longer than one month after the date.	5.4	The system needs a way to preserve a month's worth of XML files from partner agencies in case of website maintenance issues.			
5.4.2	The warehouse shall have an alert if there are problems in downloading data from partner agencies.					

DETAILED REQUIREMENTS		HIGH-LEVEL REQUIREMENTS		OPERATIONAL NEEDS		LIMITATIONS IN CURRENT SYSTEM
5.5.1	The warehouse shall automatically select loop detectors for volume counters based on proximity to the border crossing.	5.5	The system needs a way to alert administrators if volume detectors are not selected, and to allow administrators to easily select loop detectors to be used for volume counts.			
5.5.2	If a border crossing in the system does not have one or more volume loop detectors selected, it shall alert the administrator.					
5.6.1	The warehouse shall be designed in such a way that administrators can access and change the base code for the warehouse and website.	5.6	Because the code for this site will be developed with U.S. public funds, WCOG needs to retain the ability to access and change the code and allow other developers to access it.			
5.6.2	The warehouse developers shall provide documentation on the code used for the website and warehouse.					
6.1.1	The warehouse shall have an automatic alert to notify administrators when data is not received from a partner agency.	6.1	The warehouse needs a monitoring system to alert administrators when the site is down; when data packets are not received from each contributing data source; or if the backup of data fails. It also needs to create reports for evaluating performance.	6	The warehouse needs to be designed in a way that administrators can monitor the health of the warehouse and be alerted of errors.	Logs and system updates / missing data / data flags/ ftp site changes / downtime
6.1.2	The warehouse shall have an automatic alert to notify administrators when the website is down.					
6.1.3	The warehouse shall have an automatic alert to notify administrators if a backup of the data and website fails.					
6.1.4	The warehouse shall have an automatic alert to notify administrators if a new crossing or new detectors have been added to the system.					
6.1.5	The warehouse shall have an automatic alert to notify administrators if GPS data is missing for a detector.					
6.1.6	The warehouse shall have an automatic alert to notify administrators if an error occurs with the automated email system.					
6.1.7	The warehouse shall have an automatic alert to notify administrators if the server storage space reaches critical capacity.					
6.1.8	The warehouse shall have a report available for administrators that lists performance for existing detectors, crossings, data packets, and daily tasks.					

DETAILED REQUIREMENTS		HIGH-LEVEL REQUIREMENTS		OPERATIONAL NEEDS		LIMITATIONS IN CURRENT SYSTEM
6.2.1	The website shall display color coded symbols on each calendar or day view relating to the percentage of data packets received.	6.2	The warehouse needs a color-coded flagging system (i.e. green for good, yellow for caution, red for danger) for each day or month of the day/month view which would be determined by a) the number of data packets received and b) the quality of those data.			
6.2.2	The website shall display warnings to end users who run queries or analyses using data flagged as red.					
6.3.1	The warehouse shall be developed with a defined set of maintenance functions that are performed regularly to clear logs, erase backups, and continue the health of the server.	6.3	The warehouse needs an automated, scheduled clearing of logs and other files and a maintenance protocol needs to be established for long-term server health.			
6.3.2	The warehouse developers shall provide administrators with a list of daily/monthly maintenance functions and instructions on how to change that schedule or functions.					
6.4.1	The warehouse administration shall be designed in such a way that administrators can change any automatic aspect relatively easily and edit data fields.	6.4	The back-end should allow administrators to view data logs, clear logs, update FTP locations and passwords, update port names and details, and choose data counters.			
7.1.1	The warehouse shall be designed to automatically look for new ports-of-entries and detectors in each packet of data and to automatically display that new data.	7.1	The system needs to incorporate additional ports seamlessly as long as data streams abide by the established schema.	7	The warehouse needs to be built for future expansion and increased archiving requirements.	Adding new ports difficult / growing storage requirements/active lane data/comparable wait times
7.1.2	The warehouse shall be designed to alert administrators when a new port-of-entry or detector is included in the system.					
7.1.3	The warehouse shall provide steps for administrators to customize information for each new port-of-entry or detector.					
7.2.1	The warehouse shall be developed in a way that the archiving of data and the display and functioning of the	7.2	The system needs to separate the storage of			

DETAILED REQUIREMENTS		HIGH-LEVEL REQUIREMENTS		OPERATIONAL NEEDS	LIMITATIONS IN CURRENT SYSTEM
	website are separated.		archived data and the functioning of the website for greater usability.		
7.2.2	The warehouse design shall be fully documented.				
7.3.1	The warehouse hardware shall be able to store no less than 100gb of data.	7.3	The system has to be built to store increasing amount of data packets.		
7.3.2	The warehouse hardware shall be designed in such a way that additional storage capacity can be added without re-designing the warehouse structure.				
7.4.1	The warehouse hardware shall be designed to store differing types of data and different file types.	7.4	The system has to be expandable to store and display differing types of data, including but not limited to commodity data, corridor travel times, lanes open, weigh-in-motion detector data, CVISN, data, etc.		
7.4.2	The warehouse shall be designed so that new data feeds from multiple agencies and different types of equipment can be added in an incremental fashion without impacting the rest of the system.				
7.4.3	The website shall be designed in a way that allows for expanding data types across the tabbed menu bar.				
7.4.4	The website shall allow users to query the different data types available.				
7.4.5	The addition of new data shall not impact the existing data.				

Appendix D: Project Management Plan for the Cascade Gateway Border Data Warehouse

1. Purpose of Document

The purpose of this management plan is to define the scope of work, tasks, and deliverables for the Cascade Gateway Border Data Warehouse Upgrade project; to define the resources and budget available for the project; to estimate a schedule for project completion; and to describe how systems engineering will be applied to the project for validation and verification.

2. Scope of Project

The warehouse upgrade project includes the following tasks:

Task 1: Transfer existing system to a new platform and upgrade warehouse capabilities

Task 2: Develop API interface for other applications to access the warehouse

Task 3: Document development process

Task 4: Incorporate additional data components to the warehouse

More details about each task can be found in the **Concept of Operations**.

3. Budget

$150,000 is the available budget to the consulting team for completing Tasks 1-3 listed in the scope. A follow-on agreement and budget will need to be identified for any follow-on work under Task 4.

WCOG has $85,000 identified for administration of Tasks 1-4 in this portion of the overall Cascade Gateway Border Data Warehouse Upgrade & BIFA Integration project.

Additional funding under the overall project has been identified for additional data acquisition, the BIFA project architecture task, a BIFA manual, and recommendations for BIFA developers.

4. Schedule Constraints

Because this work is 50 percent funded by the Transport Canada BIFA Pilot Project Program, the project must be completed by the end of the revised agreement with Transport Canada, terminating September 30, 2011.

5. Deliverable Requirements List

The following deliverables have been identified for the project:

1. A new archive and web interface that addresses the operational needs listed above.
2. An API interface to the database.
3. Documentation on the construction and maintenance of the database and website interface.

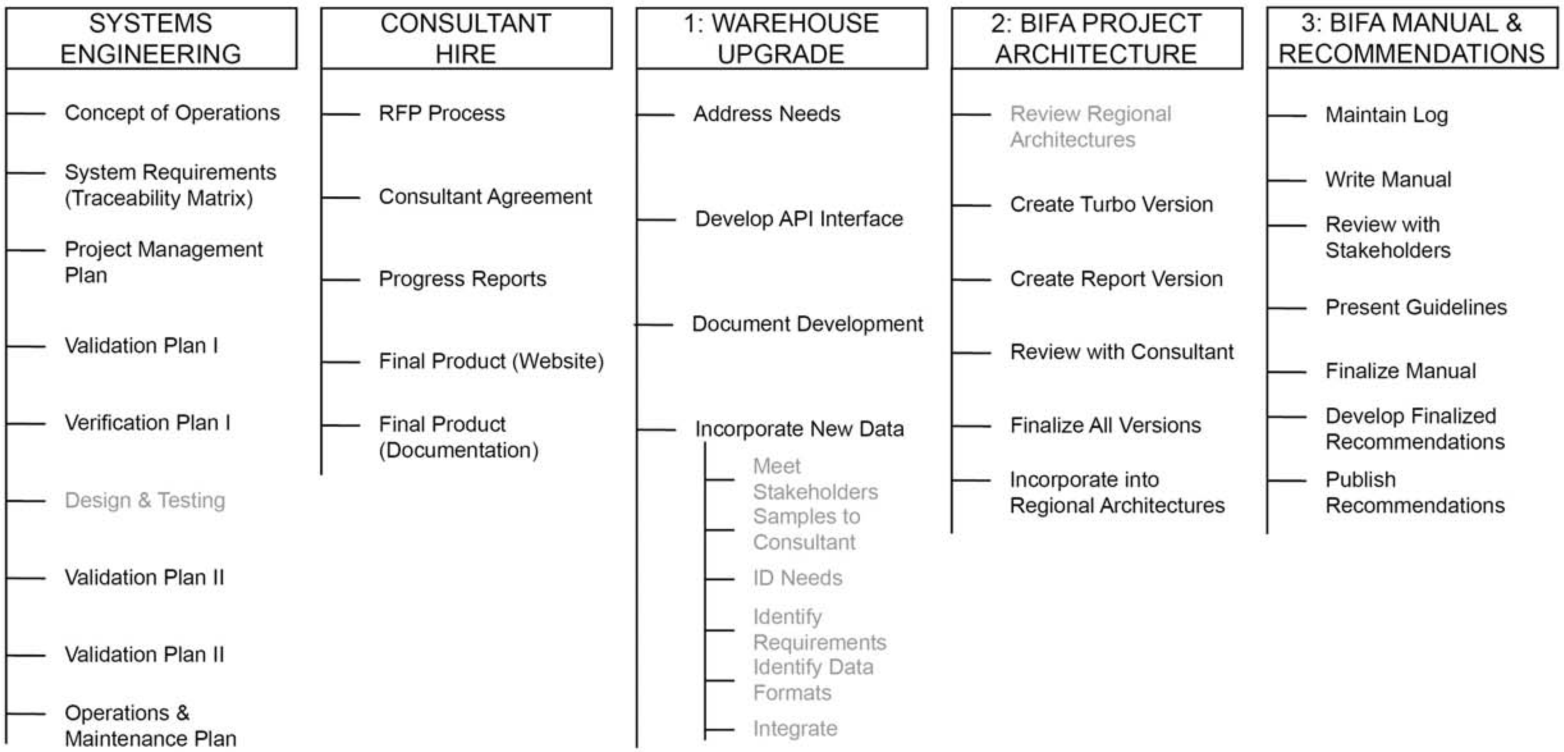
4. Possible addition of new data sets from various other data sources (to be determined separately).

Note regarding Deliverable 4: Given the unknown nature of hardware and software requirements to complete Task 4 and related Deliverable 4, this requirement pends further analysis of the systems to be implemented into the warehouse.

6. Work Breakdown Structure

See attached work breakdown structure:

CASCADE GATEWAY BORDER DATA WAREHOUSE UPGRADE & BIFA INTEGRATION PROJECT



WORK BREAKDOWN STRUCTURE
DATE: Oct. 6, 2010

Gray items are not deliverables

Appendix E: Validation Plan for the Cascade Gateway Border Data Warehouse

1. Purpose of Document

The purpose of this plan is to validate the development and deployment of the upgraded Cascade Gateway Border Data Warehouse.

The upgraded warehouse is one of three tasks under the Cascade Gateway Border Data Warehouse Upgrade and BIFA Integration project, funded by Transport Canada, U.S. Federal Highway Administration (FHWA), WA State Department of Transportation (WSDOT) and the Whatcom Council of Governments (WCOG). Therefore the validation plan applies only to the website and data warehouse improvements.

This validation plan was completed in September 2011.

2. Scope of Project

The warehouse upgrade project includes the following tasks:

Task 1: Transfer existing system to a new platform and upgrade warehouse capabilities

Task 2: Develop API interface for other applications to access the warehouse

Task 3: Document development process

Task 4: Incorporate additional data components to the warehouse

More details about each task can be found in the **Concept of Operations**.

Stakeholders

The project was administered by WCOG and overseen by an advisory team comprised of stakeholders. Agencies involved in the project included:

- B.C. Ministry of Transportation (data partnering agency, end user)
- WA State Department of Transportation (data partnering agency, funder, and end user)
- U.S. Federal Highway Administration (funder, end user)
- Transport Canada (funder, end user)
- Whatcom Council of Governments (system administrator, end user)
- University of Washington Advanced Technology Branch (data partnering agency, end user)
- End users (inspection agencies, transportation agencies, tourism organizations, trade organizations, carrier companies, cross-border shippers, customs brokers, transit agencies, researchers, and the traveling public)

3. Validation Conduct

Participants

The project was **managed and overseen by the Whatcom Council of Governments**. WCOG was responsible for all project management documentation and systems engineering. WCOG was responsible for managing the finances of the project and distributing deliverables from the project. In addition, WCOG agreed to maintain the final warehouse for five years after the completion of this project.

IBI Group, Inc. was the consulting firm selected to complete all warehouse upgrade development. Their scope of work was defined by the System Requirements, and further specified in the Project Management Plan. They were responsible for creating the upgraded data warehouse as per the specifications agreed upon from the Systems Requirements.

The **Cascade Gateway Border Data Warehouse & BIFA Integration Advisory Team** advised work on the project. The advisory team reviewed work to assure it fit the needs of each partner agency.

Location

Because this was a software development project there was no specific location. However the warehouse collects data relevant to the Cascade Gateway, a system of four border crossings between Whatcom County, Washington State and the Lower Mainland of British Columbia. These crossings and the systems which connect them represent the geographic scope of the warehouse.

Schedule

Validation occurred throughout the project, terminating at the end of the project.

The warehouse was developed and tested between November, 2010 and August 30, 2011. A final system validation occurred at the end of the project in September, 2011.

System hardware/software specifications

The system was developed as three separate parts that include the web front end (developed using ASP.NET MVC), an SQL Server 2008 database, and a windows service responsible for the parsing of XML files and sending out email notifications. The entire application was developed using Microsoft .NET 4.0 using Visual Studio 2010 Professional.

Strategy for anomalies

The following strategy for anomalies was developed at the initiation of the project. The strategy included calling an advisory team meeting to discuss problems and options for resolving the issue and secondary options for project partners to consider.

The Concept of Operations and subsequent documentation were updated to represent changes made based on any issues that arose during the process.

4. Validation Event Identification

Please see the **Systems Requirements Traceability Matrix** to reference numbers.

✓ = Pass, ✓ = Pass with caveat, X = Fail

Operational Need	Require-ments	Description	Validation Notes	Pass / Fail	Configuration Description
1. Warehouse needs to maintain existing features	1.1	Present data in similar fashion to existing site	Site layout and presentation echoes previous site.	✓	Only difference in presentation is addition of new tabs.
	1.2	Automated sign-up for reporting	Subscription tab provides visitors with this functionality	✓	The processing of XML files and sending out of notifications performed by a Windows service.
2. Present data in clear manner	2.1	Improved user guide	Help sections developed as requested; WCOG is still in the process of writing all help sections.	✓	Help available for each section of the site, and for the site as a whole.
	2.2	Incorporate new data in clear manner	New data set from BTS has clearly defined tab.	✓	New data formatted to appear like existing data sets.
	2.3	Geo-coded detector and system maps	System has high-functioning customizable abilities to map out locations of crossings and detectors.	✓	System maps use Google Maps admin interface to specify locations of crossings and detectors.
	2.4	Dynamic, customizable chart views	Google API used to develop full desired functionality.	✓	Site uses Google visualization API that depicts data in chart format.
	2.5	Easy combinations of port data	New data layouts and custom query tool allows for combining levels of data	✓	Data are organized not only by crossing but by "crossing lane" which allows for combinations of data.
	2.6	Developed as per Section 508	Site tested for Section 508 compliance by developers.	✓	Tested compliant.

Operational Need	Require-ments	Description	Validation Notes	Pass / Fail	Configuration Description
3. Provide data in downloadable, exchangeable formats on multiple platforms	3.1	Static reports	Reporting customizable by administrator	✓	Custom queries can be saved as static reports.
	3.2	Different aggregations	Aggregations of previous site plus new options	✓	Custom query allows for different aggregations of data.
	3.3	API interface	Developed as separate tab.	✓	Site includes a developer API that allows users to query warehouse and retrieve information in various formats including JSON and CSV.
	3.4	Printed and downloadable charts	In order to have the customizable chart scaling and functionality provided by Google's API we lost the ability to print charts. However downloaded data can be printed.	X	Using the Google chart function.
	3.5	Downloadable in .csv	All pages and all data downloadable.	✓	.CSV exports available on every page.
4. Quickly and efficiently pull data	4.1	Improved speed	All queries run under 30 seconds. Longer queries available only in .CSV format to expedite downloads.	✓	Database indexed to maximize performance.
	4.2	Hardware identified	New hardware expedites queries and speeds site.	✓	Improved hardware include new database server and new web server to run tasks separately.

Operational Need	Require-ments	Description	Validation Notes	Pass / Fail	Configuration Description
5. Simple to modify and maintain	5.1	Documentation of development	Documentation received.	✓	Documentation provided by IBI and noted in code.
	5.2	Accessible programming language	Developed on well-established Microsoft platform	✓	ASP.NET MVC framework
	5.3	Administrator can incorporate changes	Administrator has nearly full control of site.	✓	Much improved administrative interface and features.
	5.4	Preserve XML files from partner agencies	Given that data agency partners preserve files themselves this was later deemed unnecessary.	X	Determined unnecessary
	5.5	Alerts for administrators of site malfunctions	Warning log and detector addition alerts added.	✓	Daily warning log created for administrators and others.
	5.6	Access/permission to change programming code	All code is owned by WCOG and can be altered.	✓	Code non-proprietary.
6. Monitor health of warehouse	6.1	Monitor system for data alerts and malfunctions	Warning log and detector addition alerts added.	✓	Email alerts, detector alerts.
	6.2	Flagging system for data packet quality	Data availability color scheme added to each data set.	✓	Coded based on percentage of 288 data packets available for each day.
	6.3	Automated clearing of logs and maintenance protocol	Site developed with minimal logging and maintenance needs; ISP developed maintenance plan	✓	Server maintenance taken over by CSS Communications and is underway. Full schedule of maintenance to be developed.
	6.4	Administrators can view and clear data logs, update FTP locations and passwords, and displayed data	Administrators can edit FTP information, passwords, data. Data logs through server.	✓	Data logs stored on server not warehouse database; database components editable; server component to be discussed with maintenance schedule.

Operational Need	Require-ments	Description	Validation Notes	Pass / Fail	Configuration Description
7. Built for future expansion	7.1	Incorporate additional ports seamlessly with schema	New ports added in July.	✓	System detects new ports through schema.
	7.2	Separate storage from querying for greater speed.	Separate servers do improve speed.	✓	Separate web server and database server.
	7.3	Store increasing data.	Plenty of data storage available.	✓	600gb 15k RPM SA SCSI 6Gbps 3.5” hard drive
	7.4	Expandable to include new formats of data	New tabs can be added easily.	✓	Site designed to easily incorporate new data tables.

Appendix F: Verification Plan for the Cascade Gateway Border Data Warehouse

1. Purpose of Document

This verification document will be used to ensure that the upgraded Cascade Gateway Border Data Warehouse system meets all of the system requirements and design specifications outlined in the Concept of Operations and Validation Plan. It is the final stage for accepting system products from the archive developers.

The upgraded warehouse is one of three tasks under the Cascade Gateway Border Data Warehouse Upgrade and BIFA Integration project, funded by Transport Canada, U.S. Federal Highway Administration (FHWA), WA State Department of Transportation (WSDOT) and the Whatcom Council of Governments (WCOG). Therefore the validation plan applies only to the website and data warehouse improvements.

This verification documentation was completed in September 2011.

2. Scope of Project

The warehouse upgrade project includes the following tasks:

Task 1: Transfer existing system to a new platform and upgrade warehouse capabilities

Task 2: Develop API interface for other applications to access the warehouse

Task 3: Document development process

Task 4: Incorporate additional data components to the warehouse

More details about each task can be found in the **Concept of Operations**.

3. Verification checklist

See the spreadsheet for verification of each project element.

✓ = Pass, ✓ = Pass with caveat, X = Fail

VERIFICATION AND COMMENTS		DETAILED REQUIREMENTS	HIGH-LEVEL REQUIREMENTS	OPERATIONAL NEEDS	LIMITATIONS IN CURRENT SYSTEM
	✓	1.1.1 The data shall be presented in calendar format.	1.1 The warehouse needs to present data in a similar fashion to the existing website.	1 The warehouse needs to maintain existing features (including but not limited to port and detector data, e-mail automated reports, downloadable files, and custom queries).	Non-automated email system
	✓	1.1.2 The data shall be viewable by port-of-entry.			
	✓	1.1.3 The data shall be viewable by direction.			
	✓	1.1.4 The data shall be viewable by either detector or crossing.			
	✓	1.1.5 The data shall be queriable by custom query.			
	✓	1.2.1 The website shall allow users to sign up for automated email reports.	1.2 The warehouse needs an automated sign-up for email reporting that users can sign up for and define the thresholds for reporting themselves.		
	✓	1.2.2 The website shall allow users to define the type of reports they are interested from a set of options.			
	✓	1.2.3 The website shall allow users to unsubscribe from reports.			
	✓	2.1.1 The website shall have a help link on each page that links to information about each page.	2.1 The website needs a better user guide online and help section.	2 The warehouse needs to present data in a clear, concise manner that is easy for inexperienced end-users to find what they are looking for.	Difficult for casual end users / loop detector maps dated/ lacks dynamic charts/ not fully Section 508 compliant
	✓	2.1.2 The website administration shall allow administrators to write and edit help sections.			
	✓	2.2.1 The website shall separate out data sets using tabs at the top of the page for each data type.	2.2 The warehouse needs to incorporate the new data sets in a clear manner.		
	✓	2.2.2 The website shall develop the visual display of each data type in a similar manner.			
	✓	2.2.3 The website shall be designed in such a way that a finite number of data sets can be displayed at the top of the page.			
	✓	2.3.1 The website shall use an online mapping system to display detectors.	2.3 The warehouse needs to have geo-coded detector and system maps for users to more easily access port-level		
	✓	2.3.2 The website shall use an online mapping system to display port-of-entry crossing symbols.			

VERIFICATION AND COMMENTS		DETAILED REQUIREMENTS		HIGH-LEVEL REQUIREMENTS		OPERATIONAL NEEDS	LIMITATIONS IN CURRENT SYSTEM
	✓	2.3.3	The website mapping shall be zoomable and allow end users to pick specific detectors from the map itself.		data and individual detector data.		
	✓	2.3.4	The website shall allow administrators to input the geo-location of each detector and automatically display this detector on the map.				
	✓	2.3.5	The website shall allow administrators to edit the locations of detectors and crossings using an editing tool.				
	✓	2.4.1	Website data shall be displayed in data tables and in charts.	2.4	The website chart views needs to be dynamic and allow for end-user customization of scales, timelines, and comparisons.		
	✓	2.4.2	Website charts shall be designed in a way to allow users to customize the visual display.				
Given the 2.4.2 requirement the current set up of the site prohibits customizable charts from being printed. This is a fix for a future version since the cost of adding this feature was prohibitive.	X	2.4.3	The website chart views shall be printable.				
	✓	2.5.1	The website shall be designed in such a way that NEXUS port data and similar direction/port data can be displayed separately or together.	2.5	The website data views need to allow for easy combinations of port data (i.e. NEXUS lane data with the rest of the port data) for simpler queries.		
	✓	2.6.1	The website needs to be developed as per section 508 of the Rehabilitation Act of 1973 for accessibility for all users.	2.6	The website needs to be developed as per section 508 of the Rehabilitation Act of 1973, as amended (29 U.S.C. 794d) for accessibility for all		
	✓	2.6.2	The website developers shall instruct the administrators on how to maintain the website in such a way as to				

VERIFICATION AND COMMENTS		DETAILED REQUIREMENTS		HIGH-LEVEL REQUIREMENTS		OPERATIONAL NEEDS		LIMITATIONS IN CURRENT SYSTEM
			continue to keep all aspects of the site accessible to all users.		users.			
	✓	3.1.1	The website shall have a tab for reports.	3.1	The website should include static reports which summarize typical wait times and how a particular day compares to an average of such days. This requirement can be handled by way of "Saved Queries" for which the results can be printable.	3	The warehouse needs to provide data in downloadable and exchangeable formats that are easy to understand and capable of being used on multiple platforms.	No summary statistics/charts don't print/ No FAST/NEXUS integration
	✓	3.1.2	The reports shall be configurable and organizable by website administrators.					
	✓	3.1.3	The reports shall be viewable on screen or downloadable as .csv or equivalent file types.					
	✓	3.1.4	The website shall be able to list any number of canned reports for viewing.					
Reports are provided for a date range specified by administrator, and can range into the future.	✓	3.1.5	The reports shall be designed in such a way that they are always kept current (i.e. new data is included in the report query).					
	✓	3.2.1	The website shall include a custom query tool for end users to run unique reports on all sets of data.	3.2	The custom query tool needs to allow for different aggregations, including a daily time output (in addition to minute, hour, and am/pm outputs).			
	✓	3.2.2	The custom query shall result in multiple output types including data table, chart, and downloadable file.					
	✓	3.2.3	The custom query tool shall allow users to select date ranges and days of the week.					
	✓	3.2.4	The custom query tool shall allow users to group by month, day of week, day, am/pm, hourly, or by five minute increment.					
	✓	3.2.5	The custom query tool shall allow users to query multiple detectors or crossings at a time.					
This feature was cost-prohibitive; however administrators can develop holiday-based reports which will serve	X	3.2.6	The custom query tool shall allow users to query by holiday.					

VERIFICATION AND COMMENTS		DETAILED REQUIREMENTS		HIGH-LEVEL REQUIREMENTS		OPERATIONAL NEEDS		LIMITATIONS IN CURRENT SYSTEM
the same purpose.								
	✓	3.2.7	The custom query tool shall allow users to aggregate data by various measures, including average, max, min, and sum values.					
	✓	3.2.8	The custom query tool shall allow users to view multiple measures of data in one query (i.e. volume, wait times, etc).					
	✓	3.3.1	The warehouse shall have an associated interface to allow data to be shared with other websites.	3.3	The warehouse needs an API interface to allow the data to be "pushed" to other websites, agencies, or applications for greater dissemination of data.			
	✓	3.3.2	The warehouse API shall be designed in such a way as to maintain the security and integrity of the original database.					
See 2.4.3 comments	✓	3.4.1	The website shall produce viewable charts that can be printed; and the associated data can be downloaded in CSV format	3.4	The website charts need to be able to be printed and downloaded for end-user reports.			
	✓	3.5.1	The warehouse data shall be available in downloadable formats including but not limited to .csv format.	3.5	The warehouse data needs to be able to be downloadable in .csv format.			
	✓	4.1.1	There shall be no more than a five second delay for each page load on the website.	4.1	The warehouse access speeds need to be improved.	4	The warehouse needs to quickly and efficiently pull data from the archive.	Speed (lack thereof)
	✓	4.1.2	Large custom queries shall take no longer than ten seconds to load and display.					
	✓	4.1.3	The website shall provide a loading or waiting symbol for any query or page load that takes longer than a few seconds.					

VERIFICATION AND COMMENTS		DETAILED REQUIREMENTS		HIGH-LEVEL REQUIREMENTS		OPERATIONAL NEEDS		LIMITATIONS IN CURRENT SYSTEM
	✓	4.2.1	The warehouse shall be stored on a server with enough capacity to house exponentially increasing volumes of data.	4.2	New hardware needs to be identified to facilitate speedy data queries and increasing storage requirements.			
	✓	4.2.2	The warehouse shall be stored on a server designed in such a way that website function is separated from data storage to facilitate faster response times.					
	✓	4.2.3	The warehouse data backups shall be stored in a location that will not impact the speed or functionality of the website.					
	✓	5.1.1	The website developers shall develop documentation on how the warehouse has been developed; hardware specifications; programming knowledge needed; and how and where data are stored.	5.1	Documentation needs to be developed so the functioning of the warehouse is clear and specific for future developers.	5	The warehouse (website and data archive) needs to be developed in such a way as to make it simple to modify and maintain in the future.	Maintenance (difficult)/ files deleted
	✓	5.1.2	The website developers shall prepare documentation on how the warehouse should be maintained.					
	✓	5.1.3	The website developers shall prepare documentation on how administrators can make changes to the website.					
	✓	5.2.1	The website shall be designed using a common programming language and on a common server operating system.	5.2	Database and website formats need to be in an accessible programming language that allows for multiple developers to easily understand and address future issues.			
	✓	5.2.2	Any customized components of the warehouse software shall be described in detail in documentation with instructions on how that component ties into the full warehouse/website.					
	✓	5.3.1	The warehouse administration shall allow administrators to add or edit data feeds into the system.	5.3	System needs an easy way to incorporate changes to data			

VERIFICATION AND COMMENTS		DETAILED REQUIREMENTS		HIGH-LEVEL REQUIREMENTS		OPERATIONAL NEEDS		LIMITATIONS IN CURRENT SYSTEM
	✓	5.3.2	The warehouse administration shall allow administrators to be alerted if changes occur to the data feeds in the system.		transfer technologies or locations so that the administrator can manually change data interfaces.			
This was determined unnecessary upon further evaluation.	X	5.4.1	The warehouse shall preserve XML files downloaded from partner agencies in a backup location no longer than one month after the date.	5.4	The system needs a way to preserve a month's worth of XML files from partner agencies in case of website maintenance issues.			
	✓	5.4.2	The warehouse shall have an alert if there are problems in downloading data from partner agencies.					
This feature exists but it was later determined better to not make it automated, but rather a process overseen by the administrator.	✓	5.5.1	The warehouse shall automatically select loop detectors for volume counters based on proximity to the border crossing.	5.5	The system needs a way to alert administrators if volume detectors are not selected, and to allow administrators to easily select loop detectors to be used for volume counts.			
	✓	5.5.2	If a border crossing in the system does not have one or more volume loop detectors selected, it shall alert the administrator.					
	✓	5.6.1	The warehouse shall be designed in such a way that administrators can access and change the base code for the warehouse and website.	5.6	Because the code for this site will be developed with U.S. public funds, WCOG needs to retain the ability to access and change the code and allow other developers to access it.			
	✓	5.6.2	The warehouse developers shall provide documentation on the code used for the website and warehouse.					
	✓	6.1.1	The warehouse shall have an automatic alert to notify administrators when data is not received from a partner agency.	6.1	The warehouse needs a monitoring system to alert administrators when the site is down; when data packets are not received from each	6	The warehouse needs to be designed in a way that administrators can monitor the	Logs and system updates / missing data / data flags/ ftp site changes / downtime
This is a feature for the Internet Service Provider	✓	6.1.2	The warehouse shall have an automatic alert to notify administrators					

VERIFICATION AND COMMENTS		DETAILED REQUIREMENTS		HIGH-LEVEL REQUIREMENTS		OPERATIONAL NEEDS	LIMITATIONS IN CURRENT SYSTEM
and is being developed by them.			when the website is down.		contributing data source; or if the backup of data fails. It also needs to create reports for evaluating performance.	health of the warehouse and be alerted of errors.	
See above.	✓	6.1.3	The warehouse shall have an automatic alert to notify administrators if a backup of the data and website fails.				
	✓	6.1.4	The warehouse shall have an automatic alert to notify administrators if a new crossing or new detectors have been added to the system.				
	✓	6.1.5	The warehouse shall have an automatic alert to notify administrators if GPS data is missing for a detector.				
To simplify the email subscription system this feature was not developed.	X	6.1.6	The warehouse shall have an automatic alert to notify administrators if an error occurs with the automated email system.				
This is a feature for the Internet Service Provider and is being developed by them.	✓	6.1.7	The warehouse shall have an automatic alert to notify administrators if the server storage space reaches critical capacity.				
	✓	6.1.8	The warehouse shall have a report available for administrators that lists performance for existing detectors, crossings, data packets, and daily tasks.				
	✓	6.2.1	The website shall display color coded symbols on each calendar or day view relating to the percentage of data packets received.	6.2	The warehouse needs a color-coded flagging system (i.e. green for good, yellow for caution, red for danger) for each day or month of the day/month view which would be determined by a) the number of data packets received and b) the		
Warnings are not as specific as this but clarified in the help.	✓	6.2.2	The website shall display warnings to end users who run queries or analyses using data flagged as red.				

VERIFICATION AND COMMENTS		DETAILED REQUIREMENTS		HIGH-LEVEL REQUIREMENTS		OPERATIONAL NEEDS		LIMITATIONS IN CURRENT SYSTEM
					quality of those data.			
	✓	6.3.1	The warehouse shall be developed with a defined set of maintenance functions that are performed regularly to clear logs, erase backups, and continue the health of the server.	6.3	The warehouse needs an automated, scheduled clearing of logs and other files and a maintenance protocol needs to be established for long-term server health.			
	✓	6.3.2	The warehouse developers shall provide administrators with a list of daily/monthly maintenance functions and instructions on how to change that schedule or functions.					
	✓	6.4.1	The warehouse administration shall be designed in such a way that administrators can change any automatic aspect relatively easily and edit data fields.	6.4	The back-end should allow administrators to view data logs, clear logs, update FTP locations and passwords, update port names and details, and choose data counters.			
	✓	7.1.1	The warehouse shall be designed to automatically look for new ports-of-entries and detectors in each packet of data and to automatically display that new data.	7.1	The system needs to incorporate additional ports seamlessly as long as data streams abide by the established schema.	7	The warehouse needs to be built for future expansion and increased archiving requirements.	Adding new ports difficult / growing storage requirements/active lane data/comparable wait times
	✓	7.1.2	The warehouse shall be designed to alert administrators when a new port-of-entry or detector is included in the system.					
	✓	7.1.3	The warehouse shall provide steps for administrators to customize information for each new port-of-entry or detector.					
	✓	7.2.1	The warehouse shall be developed in a way that the archiving of data and the display and functioning of the website are separated.	7.2	The system needs to separate the storage of archived data and the functioning of the			

VERIFICATION AND COMMENTS		DETAILED REQUIREMENTS		HIGH-LEVEL REQUIREMENTS		OPERATIONAL NEEDS	LIMITATIONS IN CURRENT SYSTEM
	✓	7.2.2	The warehouse design shall be fully documented.		website for greater usability.		
	✓	7.3.1	The warehouse hardware shall be able to store no less than 100gb of data.	7.3	The system has to be built to store an exponentially increasing amount of data packets.		
	✓	7.3.2	The warehouse hardware shall be designed in such a way that additional storage capacity can be added without re-designing the warehouse structure.				
	✓	7.4.1	The warehouse hardware shall be designed to store differing types of data and different file types.	7.4	The system has to be expandable to store and display differing types of data, including but not limited to commodity data, corridor travel times, lanes open, weigh-in-motion detector data, CVISN, data, etc.		
	✓	7.4.2	The warehouse shall be designed so that new data feeds from multiple agencies and different types of equipment can be added in an incremental fashion without impacting the rest of the system.				
	✓	7.4.3	The website shall be designed in a way that allows for expanding data types across the tabbed menu bar.				
	✓	7.4.4	The website shall allow users to query the different data types available.				
	✓	7.4.5	The addition of new data shall not impact the existing data.				

Appendix G: Documentation from Website Developers

Whatcom Council of Governments

CASCADE GATEWAY DESIGN DOCUMENT

REPORT

SEP 26, 2011

IBI GROUP

IBI GROUP

IBI GROUP

1 Introduction

This document describes the overall architecture of the Cascade Data Warehouse (http://www.cascadegatewaydata.com) and is meant as an overall reference for system maintenance and any additional software development efforts. The system is composed of 3 separate parts that include the web front end (developed using ASP.NET MVC), an SQL Server 20008 database and a windows service responsible for the parsing of XML files and sending out of email notifications. The entire application is developed using Microsoft .NET 4.0 using Visual Studio 2010 Professional.

2 Overall System Architecture

The Cascade Data Warehouse was developed using ASP.NET MVC which allows for the building of a pattern based website. The Model-View-Controller (MVC) architectural pattern separates the applications into 3 components: the model, the view and the controller. ASP.NET MVC framework provides an alternative to standard ASP.NET Web Forms. The main reason for using MVC is to separate the different parts (model, view, controller) so that each part can be individually developed and tested. In general the MVC parts are responsible for:

Model – objects that are usually responsible for retrieving data from the DB and performing application level operations

View – display the application's data to the end user. The views are based on the model data

Controller – handle the user interaction, work with the model and select the view to be used to present the data.

The code throughout the entire application is well documented throughout. The code is documented at the class, interface, and method and property level. Sample documentation is provided below and the entire documentation can be exported to provide an overview document using a tool such as Doxygen.

```
/// <summary>
    /// Control for actions on the <see cref="Detector"/> object.
    /// </summary>
    public class DetectorController : Controller
    {
        /// <summary>
        /// Returns a listing of all detectors, or formatted query result for a given 
detector.
        /// GET: /Detector/
        /// GET: /Detector/{id}
        /// GET: /Detector/{id}/{year}
        /// GET: /Detector/{id}/{year}/{month}
        /// GET: /Detector/{id}/{year}/{month}/{day}
        /// </summary>
        /// <param name="id">The ID of the detector to return data for.</param>
        /// <param name="year">The year to return data for.</param>
        /// <param name="month">When provided with a year, the month to return data 
for.</param>
        /// <param name="day">When provided with a year and month, the day to return data 
for.</param>
        /// <param name="data">The data-point to create data sources for.</param>
```

```
        /// <param name="start">The date on which to start returning data,
inclusive.</param>
        /// <param name="end">The date on which to end returning data, inclusive.</param>
        /// <param name="dg">The date-level grouping to use.</param>
        /// <param name="tg">The time-level grouping to use.</param>
        /// <param name="dow">The days-of-the-week to include in the returned
data.</param>
        /// <param name="moy">The months-of-the-year to include in the returned
data.</param>
        /// <param name="format">The output format to use.</param>
        // ReSharper disable UnusedMethodReturnValue.Global
        public ActionResult Index(int? id, int? year, int? month, int? day, string start,
string end,
                                  string data = "sum-vol", string dg = null, string tg =
null, byte? dow = null,
                                  short? moy = null, string format = null)
```

The Cascade Data Warehouse MVC Application follows the MVC pattern and has the following structure within the Visual Studio 2010 Solution.

IBI.CascadeGateway

/Configuration

/Models

- AlertTypeEnumeration.cs
- CalculationEnumeration.cs
- Crossing.cs
- CrossingLane.cs
- CrossingLaneData.cs
- CrossingLaneDataQuery.cs
- CrossingMetricEnumeration.cs
- CrossingQueryDataSource.cs
- DateGroupingEnumeration.cs
- DaysOfWeekEnumeration.cs
- DelayAlertSubscription.cs
- Detector.cs
- DetectorData.cs
- DetectorMetricEnumeration.cs
- DetectorQueryDataSource.cs
- DirectionOfTravelEnumeration.cs
- Feed.cs
- FeedData.cs

  - FeedStatus.cs
  - FeedStatusCodeEnumeration.cs
  - GpsData.cs
  - HelpArticle.cs
  - IQueryDatasource.cs
  - LaneTypeEnumeration.cs
  - LogEntry.cs
  - MonthsOfYearEnumeration.cs
  - OrganizationIdentifier.cs
  - OutputFormatEnumeration.cs
  - Query.cs
  - QueryResult.cs
  - TimeGroupingEnumeration.cs

/Repositories

  - /SQLServer/
    - BtsQueryRepository.cs
    - CrossingLaneRepository.cs
    - CrossingRepository.cs
    - DelayAlertRepository.cs
    - DetectorRepository.cs
    - FeedRepository.cs
    - GpsDataRepository.cs
    - HelpArticleRepository.cs
    - ISqlServerRepositoryContext.cs
    - LogRepository.cs
    - QueryRepository.cs
    - TimestampRepository.cs
  - IBtsQueryRepository.cs
  - ICrossingLaneRepository.cs
  - ICrossingRepository.cs
  - IDelayAlertRepository.cs
  - IDetectorRepository.cs
  - IFeedRepository.cs
  - IGpsDataRepository.cs

- IHelpArticleRepository.cs
- ILogRepository.cs
- IQueryRepository.cs
- IRepositoryContext.cs
- ITimestampeRepository.cs
- ITransaction.cs
- RepositoryContext.cs
- RepositoryContextFactory.cs
- Transaction.cs

IBI.CascadeGateway.MvcApplication

/Content

- /Images – all images used throughout the application

/Controllers

- AccountController.cs
- CrossingController.cs
- CrossingLaneController.cs
- CustomQueryController.cs
- DetectorController.cs
- FeedController.cs
- FreightController.cs
- HelpArticleController.cs
- HomeController.cs
- SubscriptionController.cs

/Models (contains all the models used throughout the application

- AccountModels.cs
- ChartViewModels.cs
- CrossingEditViewModels.cs
- CrossingIndexViewModel.cs
- CustomQueryViewModel.cs
- DetectorEditViewModel.cs
- DetectorIndexViewModel.cs
- DetectorListViewModel.cs
- FeedEditViewModel.cs
- FreightIndexViewModel.cs

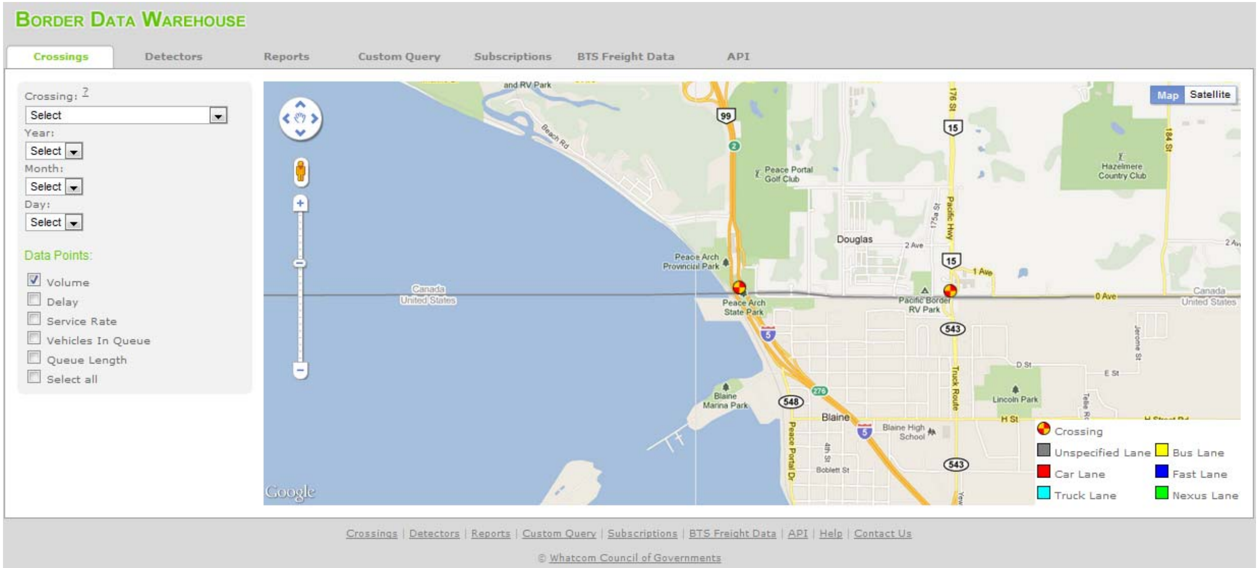
  - o HelpArticleEditViewModel.cs
  - o IndexViewModel.cs
  - o SubscriptionEditViewModel.cs
- /Scripts (contains scripts such as AJAX and JQUERY)
- /Views
  - o /Account
  - o /Crossing
  - o /CrossingLane
  - o /CustomQuery
  - o /Detector
  - o /Feed
  - o /Freight
  - o /HelpArticle
  - o /Home
  - o /Shared
  - o /Subscription

3 Front End (Website)

The following section describes the Web Front End for the Cascade Gateway Data (www.cascadegatewaydata.org). As was described above the site uses ASP.NET MVC to create the website.

3.1 Crossings

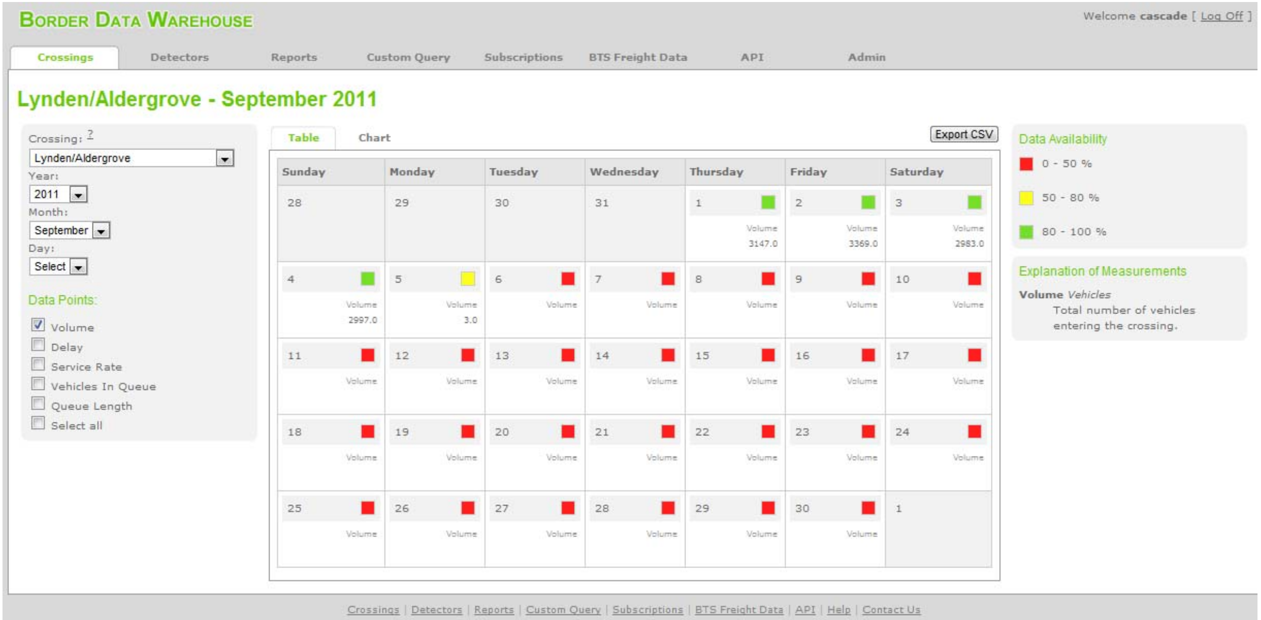
The Crossings Page displays the currently defined crossings and crossing lanes. The Crossing Lanes are available at lower zoom levels and are depicted on the map using the icons presented in the legend in the screenshot below. The screen is used to navigate to the crossing data views through the left hand navigation. The user can select the Crossing, Year, Month, Day as well as any number of Data Points.



Crossing Screenshot

3.2 Crossing Data – Calendar View

The Crossing Data Calendar View depicts the data points for a selected crossing using a calendar view for either the annual or month view. The calendar view displays the selected data points as well as the data availability for the selected month or day. The data availability is calculated based on the percentage of all the records for a specific time period. For a specific day there are 288 total values and the percentage is based on how many of those values were collected.



Crossing Calendar View

3.3 Crossing Data – Table View

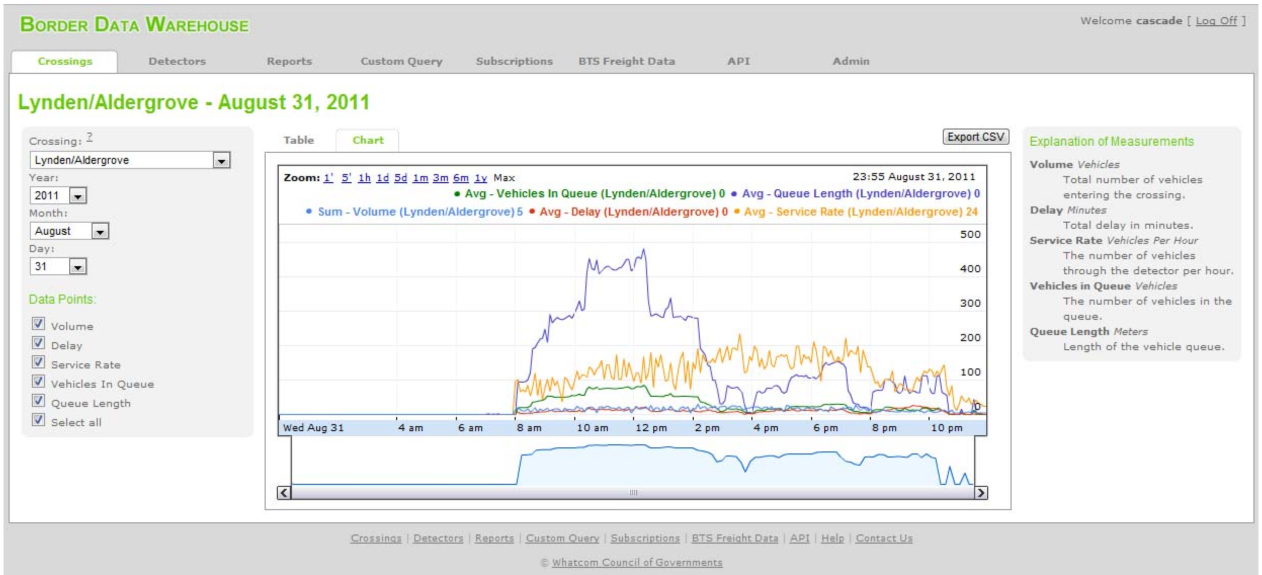
The Crossing Data Table View is used to depict crossing data for a single day based on the user's selection of a crossing, or a crossing lane, year, month, day and data points. The data is depicted in table with alternating rows and options to view the data in a chart as well as export to CSV. The data is sorted in ascending order by date.

The table distinguishes between empty values and 0s as can be seen in the screenshot below. If a specific cell is empty it means that no data was provided for the specific time slot.

Crossing Data View

3.4 Crossing Data Chart View

The Crossing Data Chart View represents the same data as the Table View, but it uses the Google visualization API and depicts the data in form of a chart. The chart has the timeframe along the X axis with the selected data points along the Y axis. The chart also allows the user to change the timeframe by shrinking it using the slider below the chart or changing it via the provided Zoom Levels (1', 5', 1h, etc). The chart also allows the user to interact with the chart by hovering with the mouse over any data point.



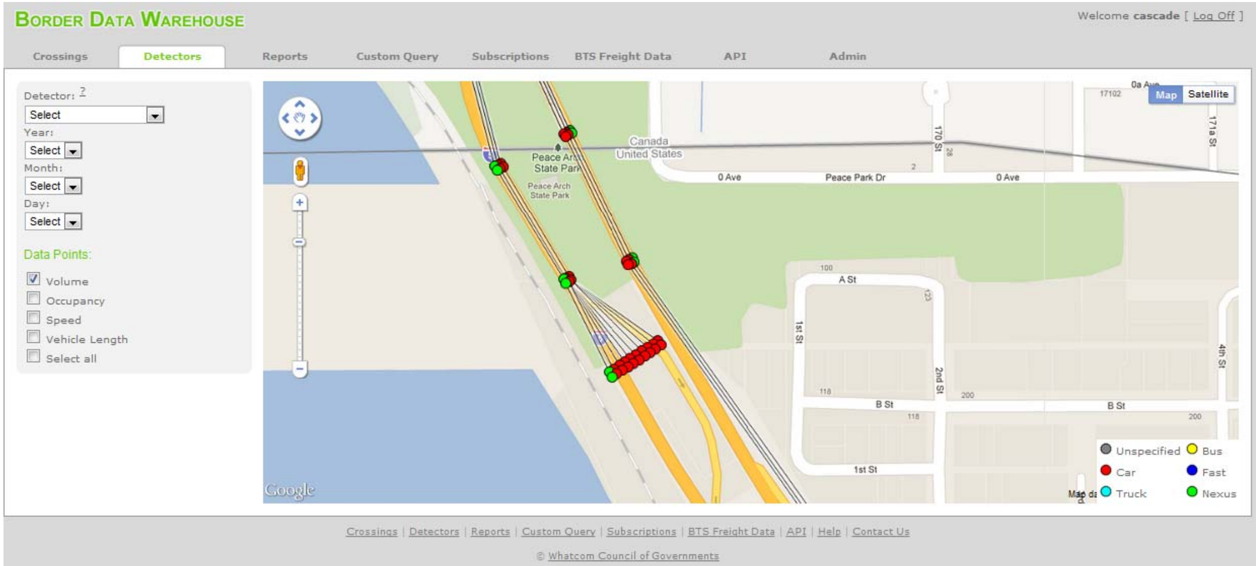
Crossing Chart View

3.5 Crossing Data - CSV Export

The Crossing Data CSV Export allows the selected data to be exported to CSV. The created file is called query.csv and it will include the data points selected by the user, with the first data row being the column definition.

3.6 Detectors

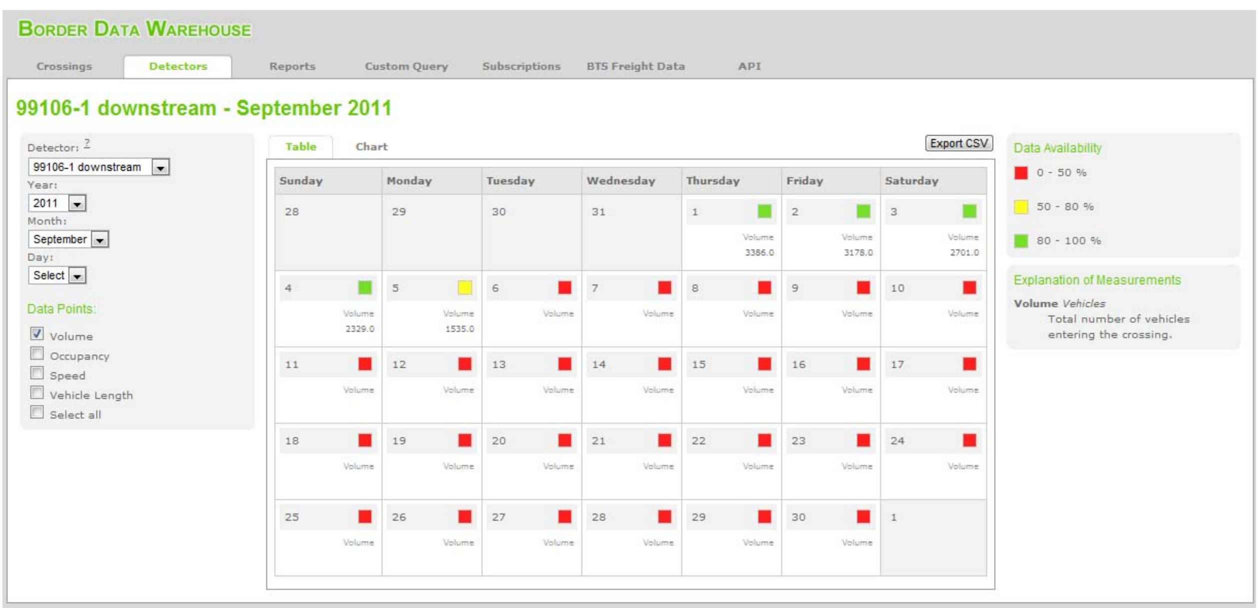
The Detector map view is similar to the crossing map view. It displays the detectors including the connectivity of each detector to its upstream detector (if configured). The user can navigate to the detector data by selection the detector, year, month, day and any of the detector Data Points.



Detector Map View

3.7 Detector Data – Calendar View

The Detector data Calendar View displays information for a particular detector using a calendar in either an annual or a month layout depending the data selected by the user. Each calendar cell can be clicked by the user to access data for a particular day. The calendar also features the availability of data which is calculated based on the 288 data samples per day.



Detector Calendar View

3.8 Detector Data – Table View

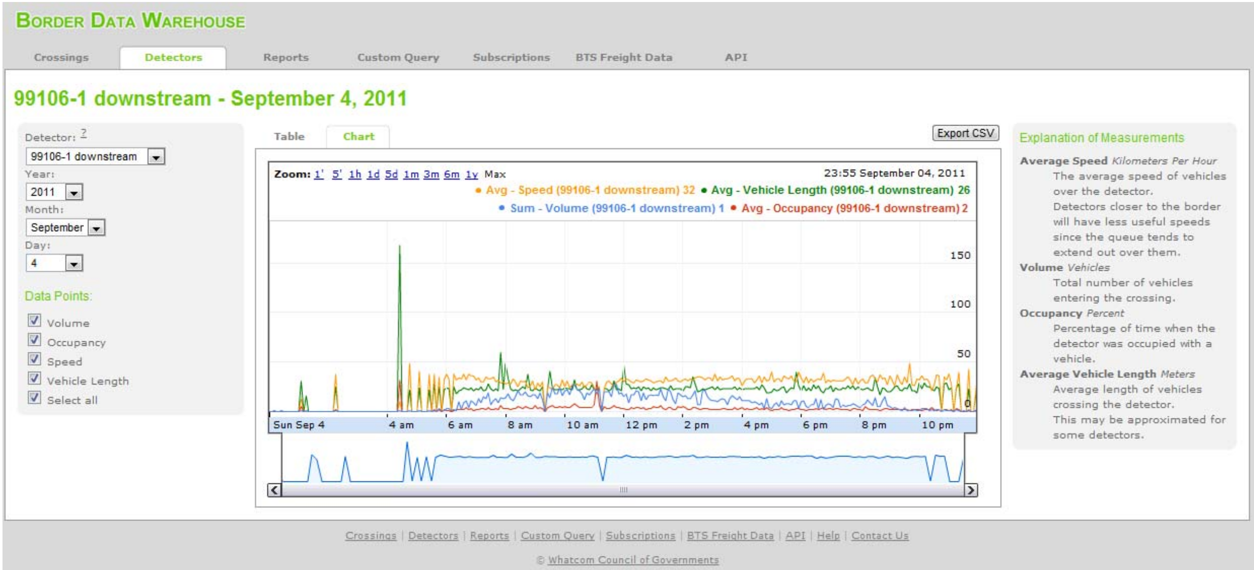
The Detector Data Table View allows the user to view data for a specific detector on a specific day. The data is sorted by the date & time (descending) and allows quick access to the chart view and the data export.

Time	Sum - Volume	Avg - Occupancy	Avg - Speed	Avg - Vehicle Length
Sep 04 00:00	0.0	0.0	0.0	0.0
Sep 04 00:05	0.0	0.0	0.0	0.0
Sep 04 00:10	1.0	0.0	0.0	0.0
Sep 04 00:15	0.0	0.0	0.0	0.0
Sep 04 00:20	0.0	0.0	0.0	0.0
Sep 04 00:25	1.0	0.0	0.0	0.0
Sep 04 00:30	0.0	0.0	0.0	0.0
Sep 04 00:35	0.0	0.0	0.0	0.0
Sep 04 00:40	0.0	0.0	0.0	0.0
Sep 04 00:45	0.0	0.0	0.0	0.0
Sep 04 00:50	0.0	0.0	0.0	0.0
Sep 04 00:55	0.0	0.0	0.0	0.0
Sep 04 01:00	0.0	0.0	0.0	0.0
Sep 04 01:05	1.0	5.0	13.0	31.0
Sep 04 01:10	0.0	0.0	0.0	0.0
Sep 04 01:15	1.0	1.0	0.0	16.0
Sep 04 01:20	0.0	0.0	0.0	0.0
Sep 04 01:25	0.0	0.0	0.0	0.0
Sep 04 01:30	0.0	0.0	0.0	0.0
Sep 04 01:35	0.0	0.0	0.0	0.0
Sep 04 01:40	0.0	0.0	0.0	0.0

Detector Data View

3.9 Detector Data – Chart View

The Detector Data Chart View represents the same data as the Table View, but it uses the Google visualization API and depicts the data in form of a chart. The chart has the timeframe along the X axis with the selected data points along the Y axis. The chart also allows the user to change the timeframe by shrinking it using the slider below the chart or changing it via the provided Zoom Levels (1’, 5’, 1h, etc). The chart also allows the user to interact with the chart by hovering with the mouse over any data point.



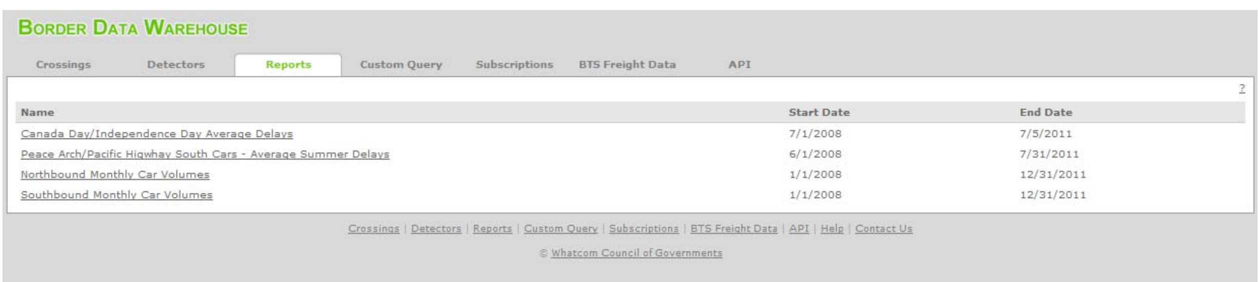
Detector Chart View

3.10 Detector Data – CSV Export

The Detector Data CSV Export allows the selected data to be exported to CSV. The created file is called query.csv and it will include the data points selected by the user, with the first data row being the column definition.

3.11 Reports

The Reports area allows the public users quick access to saved reports. Reports can be created and made available to the public by an administrator. Each of the reports is a saved custom query. The user can click on the link and view the results the same way as the custom query.



Report View

3.12 Custom Query

The Custom Query capability allows users of the site to create complicated queries for both detector and crossing data. The following describes the input parameters for the custom query:

> Start / End Date – the start/end date selectors allow for the selection of the time interval for the query.

Days of Week Filter – the filter allows only certain days to be included in the results (default – all selected).

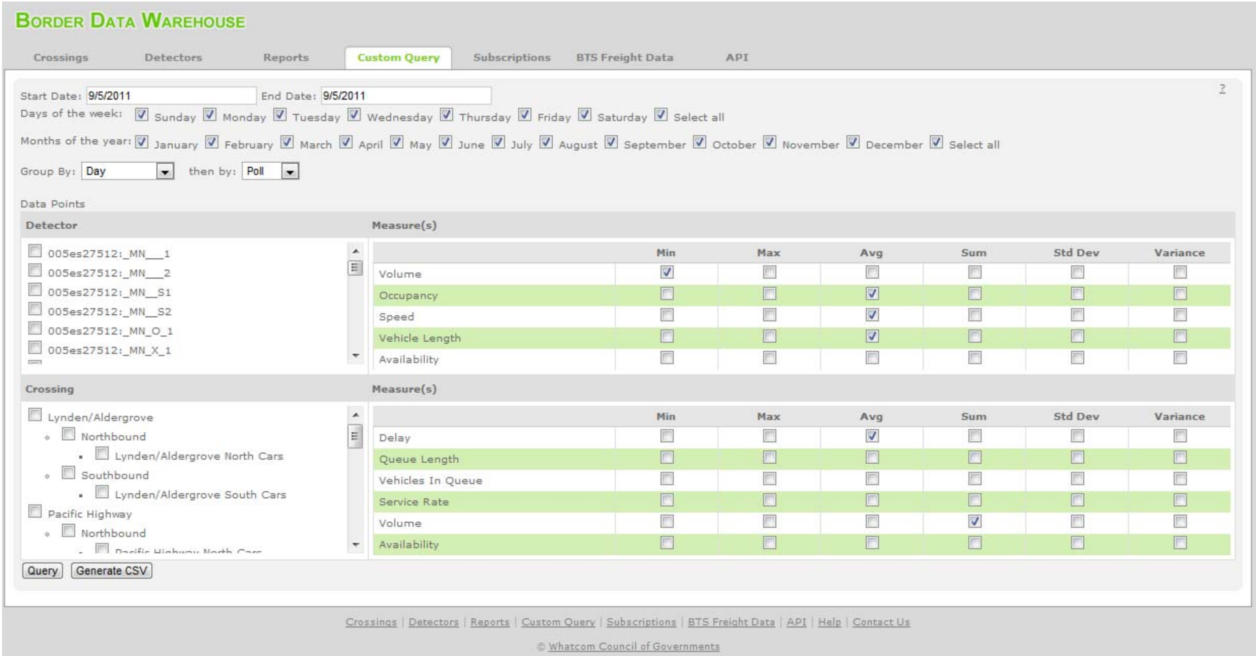
Months of Year Filter – the filter allows only certain months to be included in the results (default – all selected).

Group By – the Group By aggregation allows the user to group by Day, Day of Week, Month, Month of Year, Year.

Than By – the Than By aggregation allows for a sub grouping within the Group By clause, allowing for Poll (5 min data), Hour, AM/PM, and Day.

Detector Data Points – the selection allows for any number of detectors and for each detector any combination of data points (volume, occupancy, speed, vehicle length, availability) and calculations (min, max, average, sum, standard deviation and variance).

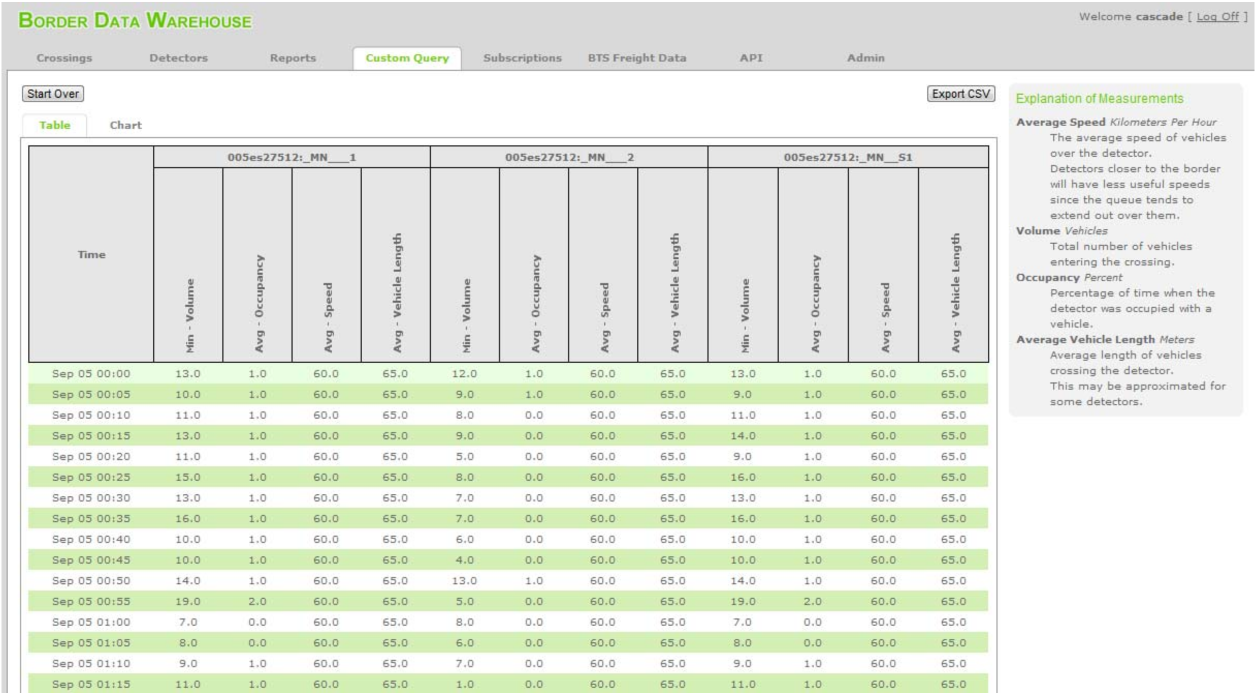
Crossing Data Points – the selection allows for any number of crossings and for each crossing any combination of data points (delay, queue length, vehicles in queue, service rate, volume, availability) and calculations (min, max, average, sum, standard deviation and variance).



Custom Query

3.13 Custom Query – Table View

The standard output for the Custom Query is the Table View. The table view uses vertical column headings in order to display all the selected data points for both the crossings and detectors. As can be seen from the screenshot below, the data points are grouped within each detector or crossing. The data is sorted by date & time and can be viewed as a chart and exported to CSV.



Time	005es27512:_MN___1				005es27512:_MN___2				005es27512:_MN__S1			
	Min - Volume	Avg - Occupancy	Avg - Speed	Avg - Vehicle Length	Min - Volume	Avg - Occupancy	Avg - Speed	Avg - Vehicle Length	Min - Volume	Avg - Occupancy	Avg - Speed	Avg - Vehicle Length
Sep 05 00:00	13.0	1.0	60.0	65.0	12.0	1.0	60.0	65.0	13.0	1.0	60.0	65.0
Sep 05 00:05	10.0	1.0	60.0	65.0	9.0	1.0	60.0	65.0	9.0	1.0	60.0	65.0
Sep 05 00:10	11.0	1.0	60.0	65.0	8.0	0.0	60.0	65.0	11.0	1.0	60.0	65.0
Sep 05 00:15	13.0	1.0	60.0	65.0	9.0	0.0	60.0	65.0	14.0	1.0	60.0	65.0
Sep 05 00:20	11.0	1.0	60.0	65.0	5.0	0.0	60.0	65.0	9.0	1.0	60.0	65.0
Sep 05 00:25	15.0	1.0	60.0	65.0	8.0	0.0	60.0	65.0	16.0	1.0	60.0	65.0
Sep 05 00:30	13.0	1.0	60.0	65.0	7.0	0.0	60.0	65.0	13.0	1.0	60.0	65.0
Sep 05 00:35	16.0	1.0	60.0	65.0	7.0	0.0	60.0	65.0	16.0	1.0	60.0	65.0
Sep 05 00:40	10.0	1.0	60.0	65.0	6.0	0.0	60.0	65.0	10.0	1.0	60.0	65.0
Sep 05 00:45	10.0	1.0	60.0	65.0	4.0	0.0	60.0	65.0	10.0	1.0	60.0	65.0
Sep 05 00:50	14.0	1.0	60.0	65.0	13.0	1.0	60.0	65.0	14.0	1.0	60.0	65.0
Sep 05 00:55	19.0	2.0	60.0	65.0	5.0	0.0	60.0	65.0	19.0	2.0	60.0	65.0
Sep 05 01:00	7.0	0.0	60.0	65.0	8.0	0.0	60.0	65.0	7.0	0.0	60.0	65.0
Sep 05 01:05	8.0	0.0	60.0	65.0	6.0	0.0	60.0	65.0	8.0	0.0	60.0	65.0
Sep 05 01:10	9.0	1.0	60.0	65.0	7.0	0.0	60.0	65.0	9.0	1.0	60.0	65.0
Sep 05 01:15	11.0	1.0	60.0	65.0	1.0	0.0	60.0	65.0	11.0	1.0	60.0	65.0

Custom Query Data View

3.14 Subscriptions

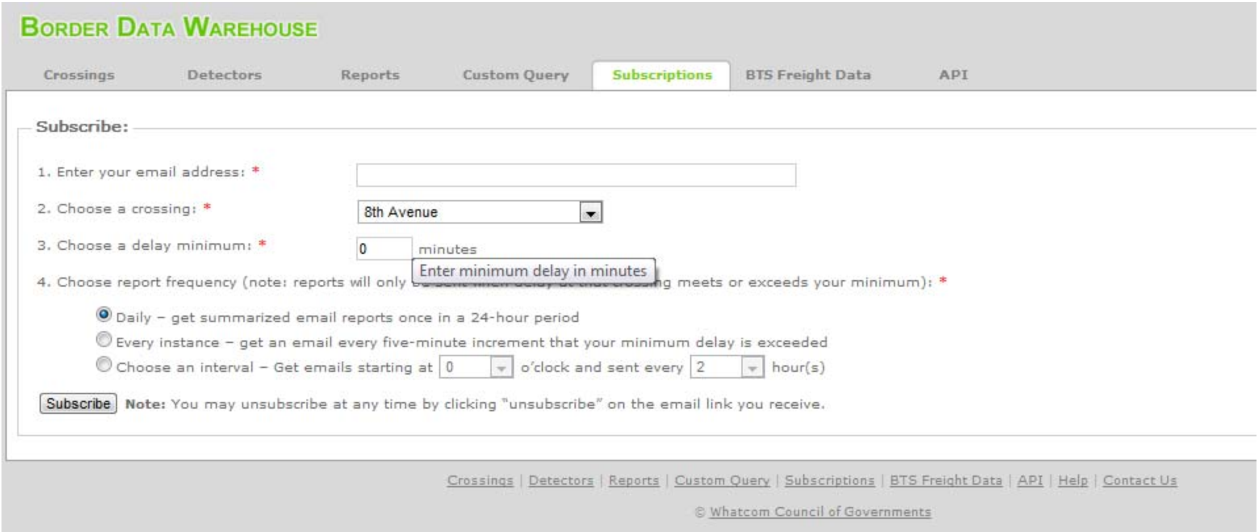
The Subscription page allows for the public users to receive notification reports via email. The reports are available to the users for any crossing where the delay at the crossing exceeds a value defined by the user. The user also has the ability to choose from 3 notification types as can be seen in the screenshot below. These notification types include:

Daily – receive a daily email for a specific Crossing where the delay exceeded the minimum threshold.

Every Instance – receive an email every time the delay exceeds the minimum threshold (real time).

Choose an Interval – receive an email based on an interval (hourly) every time the delay exceeds the minimum threshold.

The notification emails formats are described in the Subscription Section below.



Subscriptions

3.15 BTS Freight Data

The BTS Freight Data screen allows the public users to query the BTS Freight Data System. The cascade data warehouse acts as a query front end which dispatches the calls to the system and displays the information in a manner uniform with the rest of the site.

The BTS Freight query allows the user to select the following input parameters:

Year – year for the query

Measure – Value or Weight

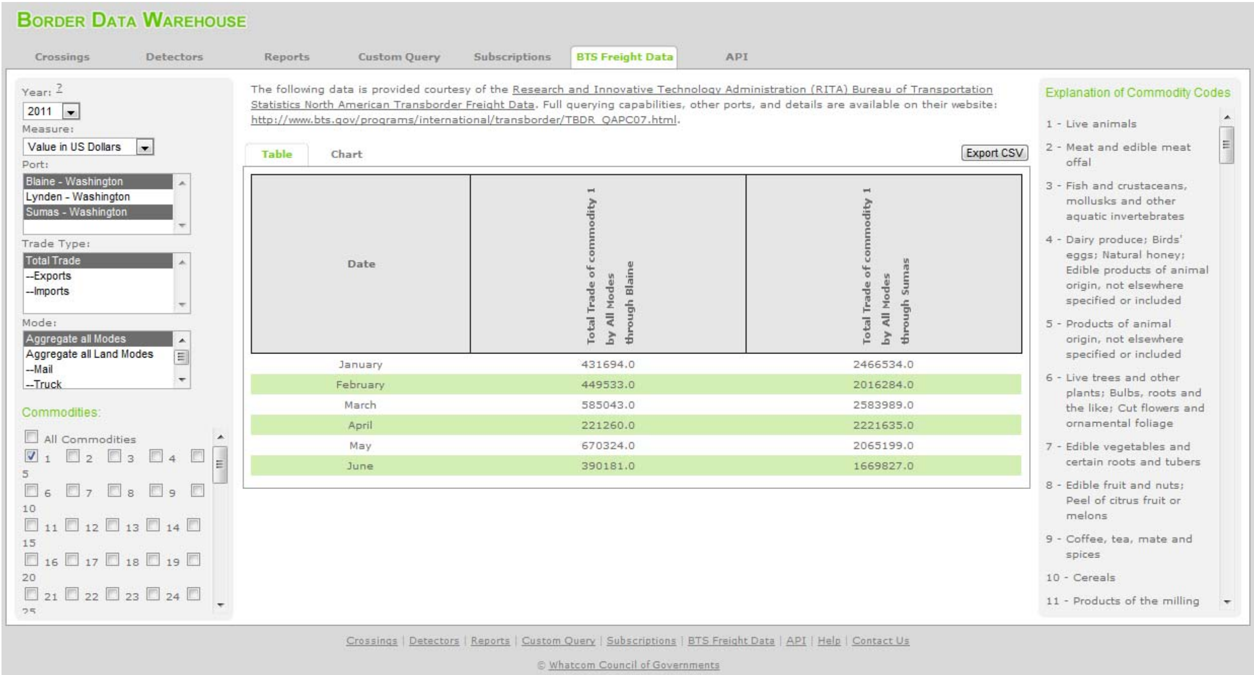
Port – allow the user to select multiple values from Blaine, Lynden and Sumas

Trade Type – Both, Export, Import

Mode – allow the user to select from any of the predefined land, air and sea modes

Commodities – allow the user to select any number of commodity codes.

The query that is performed by the BTS Freight Data System is displayed using the standard Table, Chart and Export Views providing a consistent data output across the entire site.



BTS Freight Query

3.16 API

The API page provides an overview of the Cascade Data Warehouse Developer API. The page acts as a resource to any developer that would like information regarding how the warehouse can be queried for data. The API reference is example based providing the users quick and easy to understand examples about how queries can be formed and executed.

3.17 Login

The Login page for the site is available at www.cascadegatewaydata.org/admin. The login secures the administration area from the public part of the site. The validation of the username and password is performed against the database where the username and password are securely stored.

3.18 Crossing Administration

The Crossing Administration allows the administrator to view each configured crossing in the system. The crossings are presented in a list and can be individually edited. Each crossing contains the following information:

Name – display name for the crossing.

Canadian Roadway Name – roadway name leading to/from the crossing on the Canadian side of the border.

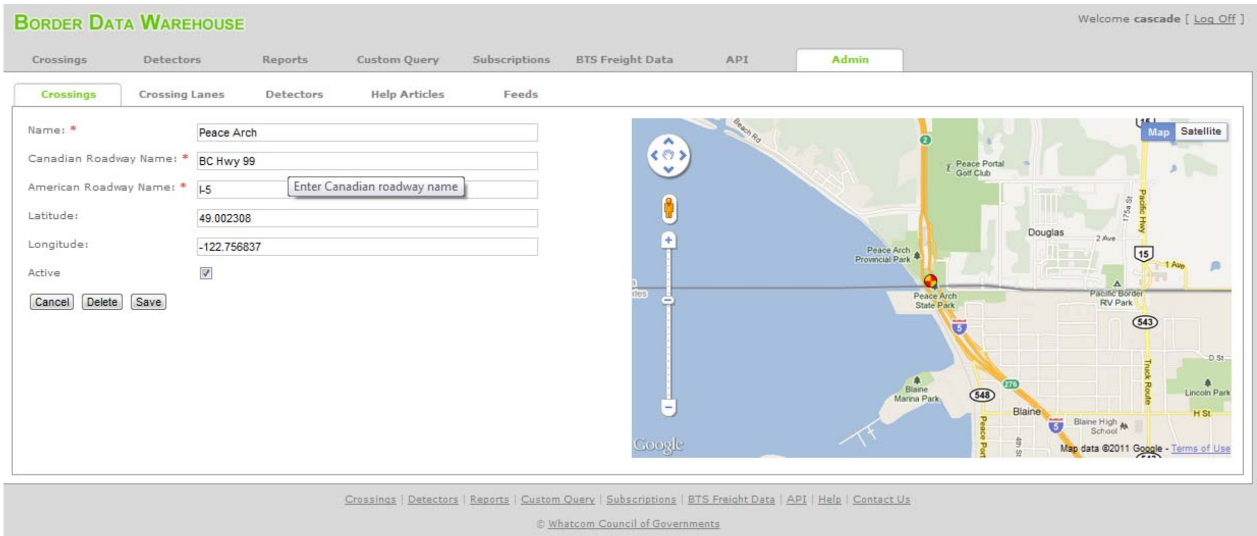
US Roadway Name - roadway name leading to/from the crossing on the US side of the border.

Latitude – latitude for the crossing (display)

Longitude – longitude for the crossing (display)

Active – whether the crossing is active. Marking a crossing inactive will disable it from view on the entire site including the maps and queries.

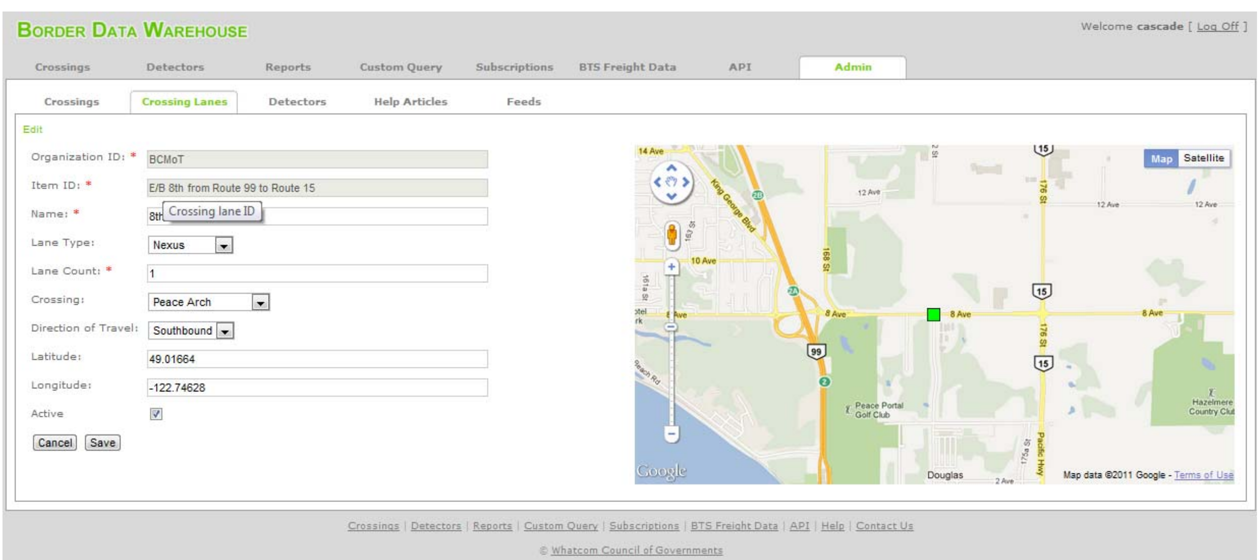
Apart from the properties, the user can also use the map to configure the location of the crossing. The icon on the map can be dragged-and-dropped to a desired location. The latitude and longitude fields are automatically updated as the user moves the icon.



Crossing Administration

3.19 Crossing Lane Administration

The Crossing Lane Administration allows the user to edit individual crossing lanes for a specific crossing and direction. The administration uses the same approach as the crossing administration allowing the user to populate the data fields and also to select the exact location of the crossing lane by dragging the icon to the appropriate location on the map.

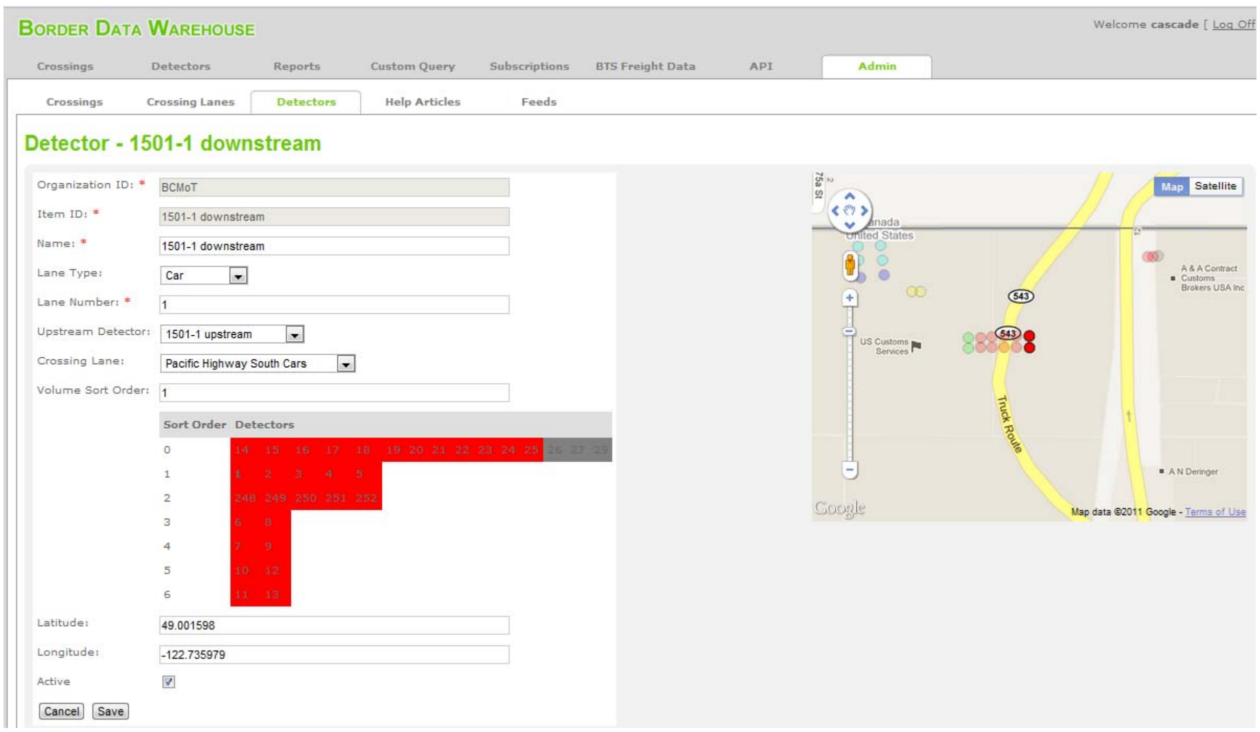


Crossing Lane Administration

3.20 Detector Administration

Detector Administration is used to place the detector on the map and to also map the detector to a specific crossing lane and to also connect a detector with its upstream detector. The selection of a detector to a particular crossing lane is important since a detector can contribute to the volume for a particular crossing lane. With the use of a sort order the detectors can be arranged into rows that contribute to the volume for a crossing lane. As depicted in the screenshot below, the detector being edited contributes volume to the Pacific Highway South Cars with a volume sort order of 1. This assignment means that in most cases the volume calculation is performed by volume sort order 0 (first row, closest to the crossing), but if any of the detectors are failed the volume will be calculated by the next row.

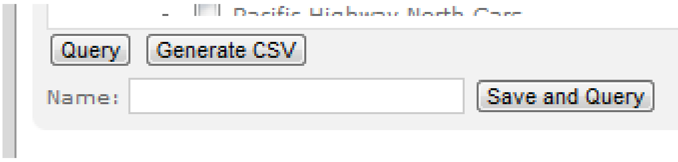
The map gives a quick overview of the other detectors in the area (greyed out) and allows for the positioning of the detector with drag-and-drop actions on the map.



Detector Administration

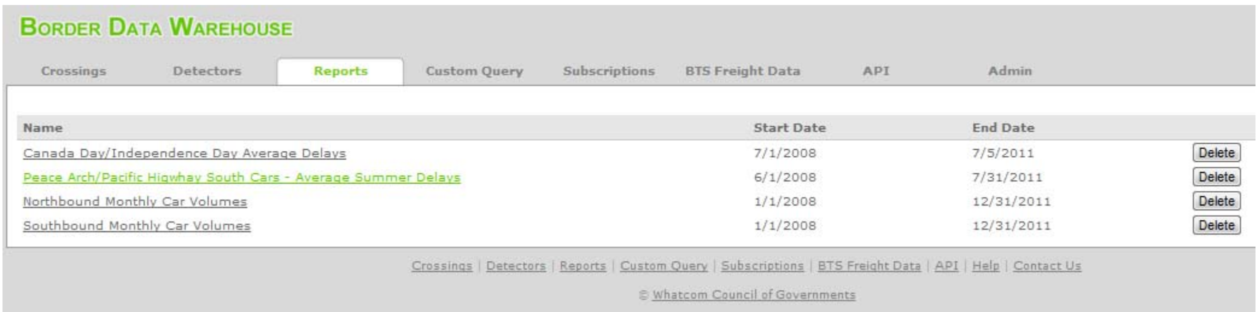
3.21 Reports Administration

The administration of save Reports requires the administrator to save and name a custom query. The query will be made available to the public users.



Reports Administration

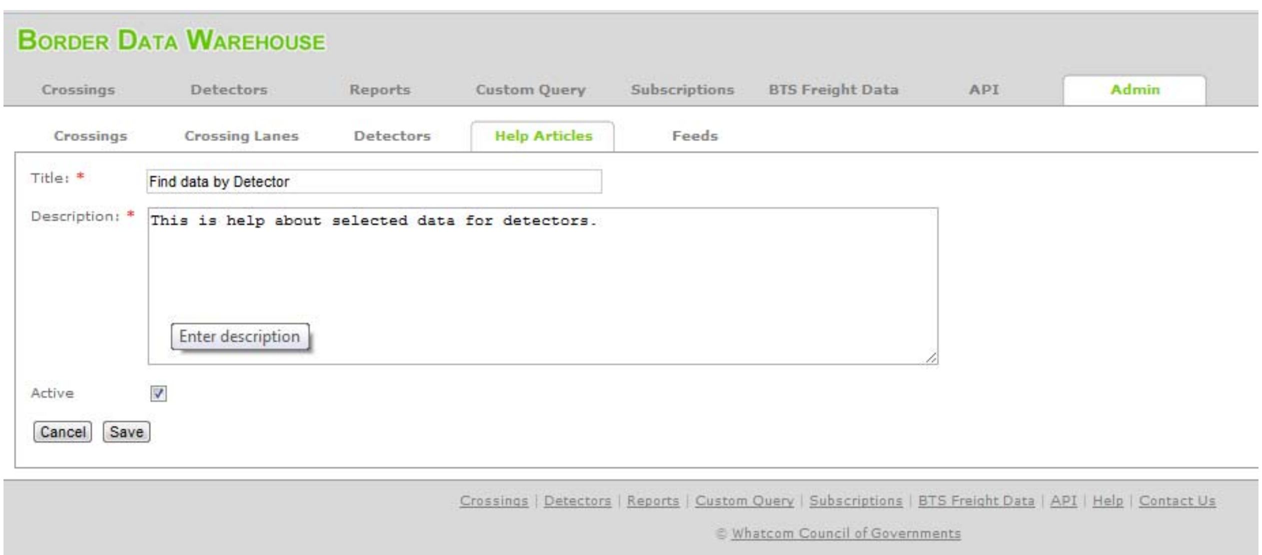
Existing reports can be deleted from the site by the administrator by clicking delete (when logged in)



Reports View

3.22 Help Article Administration

Help articles appear throughout the site and are depicted using the ? symbol. Each of the help symbols has been placed on a page and is associated with a help article that can be managed by the administrator. Each article has a title and a description that is visible to the public users when he/she hovers over the ? symbol.



Help Article Administration

3.23 Feed Administration

The warehouse collects its data from 2 data feeds that provide detector and crossing data via a standard XML file. The feed administration capability allows for the disabling of the feeds (Active flag) as well as the changing of the format of the URL. For each feed the URL provides the means of accessing the data from the data source. The URL contains the following syntax that is used to specify that file naming convention used by the data source:

> http://www.th.gov.bc.ca/bchighwaycam/r-atis/atis/border/mot_atis_{0:yyyyMMdd-HHmm}.xml - as can be seen from the sample URL, the fixed part of the feed contains the location of the file and the start of the file name. {0:yyyyMMdd-HHmm} depicts the changing part of the filename that describes the date & time for each file.

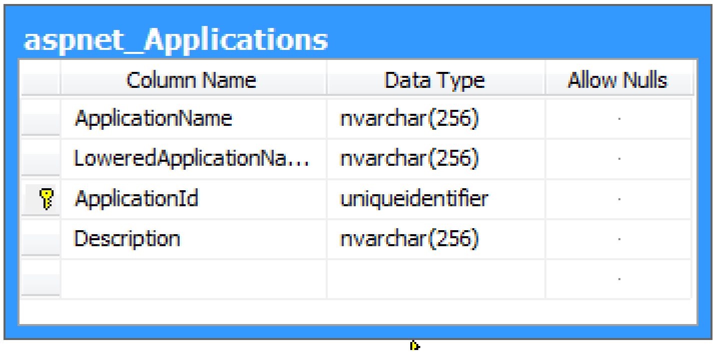
Feed Administration

4 Database Structure

The following sections describe the structure of the Cascade Data Warehouse Database. The database that houses the data for the warehouse is SQL Server 2008. Each section outlines the table including an explanation of all fields, indexes and the uses of the data.

4.1 aspnet_Applications

The aspnet_Application table is used to store the applications for the purpose of the Membership and Role providers. The applicationID is referenced by the other membership data tables.



aspnet_Applications

	Column Name	Data Type	Allow Nulls
	ApplicationName	nvarchar(256)	
	LoweredApplicationNa...	nvarchar(256)	
🔑	ApplicationId	uniqueidentifier	
	Description	nvarchar(256)	

Field Name	Type	Description
ApplicationName	String (256)	Name of the application
LoweredApplicationName	String (256)	Name of the application in lower case
ApplicationId (PK)	ID	Unique Identifier for the application
Description	String (256)	Description of the application.

4.2 aspnet_Membership

The aspnet_Membership table stores the users & passwords used to access the application. Each user is associated with an application and the stored usernames, passwords and the status of the account such as whether the account is locked and the last time the account was accessed. Sensitive information such as the username and passwords are encrypted.

aspnet_Membership

	Column Name	Data Type	Allow Nulls
	ApplicationId	aspnet_Membership fier	
🔑	UserId	uniqueidentifier	
	Password	nvarchar(128)	
	PasswordFormat	int	
	PasswordSalt	nvarchar(128)	
	MobilePIN	nvarchar(16)	
	Email	nvarchar(256)	
	LoweredEmail	nvarchar(256)	
	PasswordQuestion	nvarchar(256)	
	PasswordAnswer	nvarchar(128)	
	IsApproved	bit	
	IsLockedOut	bit	
	CreateDate	datetime	
	LastLoginDate	datetime	
	LastPasswordChange...	datetime	
	LastLockoutDate	datetime	
	FailedPasswordAttem...	int	
	FailedPasswordAttem...	datetime	
	FailedPasswordAnswe...	int	
	FailedPasswordAnswe...	datetime	
	Comment	ntext	

Field Name	Type	Description
ApplicationID	ID	Unique ID for the application
UserID (PK)	ID	Unique ID for the user
Password	String (128)	Password (plaintext, hashed, or encrypted; base-64-encoded if hashed or encrypted)
PasswordFormat	Int	Password format (**0**=Plaintext, **1**=Hashed, **2**=Encrypted)
PasswordSalt	String (128)	Randomly generated 128-bit value used to salt password hashes; stored in base-64-

Field Name	Type	Description
		encoded form
MobilePIN	Stirng (16)	User's mobile PIN (currently not used)
Email	String (256)	User's email address
LoweredEmail	String (256)	User's email address in lowercase
PasswordQuestion	String (256)	Password question
PasswordAnswer	String (256)	Password answer
IsApproved	Boolean	Whether the account is approved
IsLockedOut	Boolean	Whether the account is locked
CreateDate	Date & Time	Date & Time the account was created
LastLoginDate	Date & Time	Date & Time of the last successful login
LastPasswordChangeDate	Date & Time	Date & Time when the password was changed
LastLockoutDate	Date & Time	Date & Time when the account was locked
FailedPasswordAttemptCount	Int	Number of consecutive failed login attempts
FailedPasswordAttempt-WindowStart	Date & Time	Date and time of first failed login if FailedPasswordAttemptCount is nonzero
FailedPasswordAnswer-AttemptCount	Int	Number of consecutive failed password answer attempts
FailedPasswordAnswer-AttemptWindowStart	Date & Time	FailedPasswordAnswerAttemptCount is nonzero
Comment	String	Additional comments

4.3 aspnet_Roles

The aspnet_Roles table defines the configured roles in the system. The data warehouse defines a single role: Administrator.

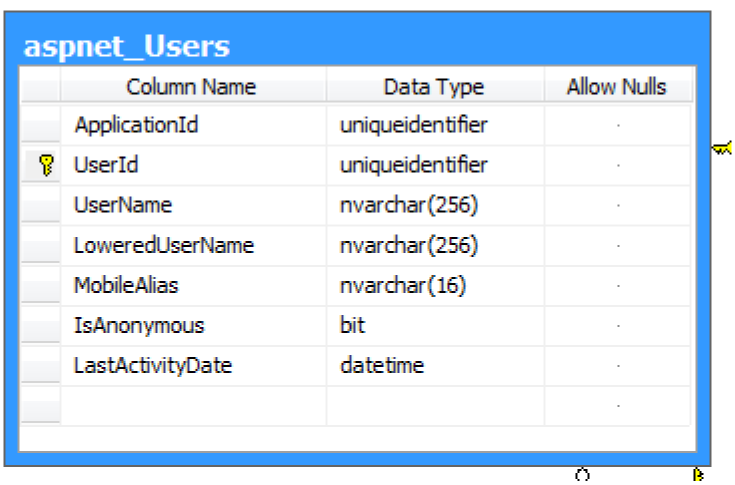
aspnet_Roles

Column Name	Data Type	Allow Nulls
ApplicationId	uniqueidentifier	
RoleId	uniqueidentifier	
RoleName	nvarchar(256)	
LoweredRoleName	nvarchar(256)	
Description	nvarchar(256)	

Field Name	Type	Description
ApplicationId	ID	Unique ID for the application
RoleId (PK)	ID	Unique ID for the role
RoleName	String (256)	Name of the role.
LoweredRoleName	String (256)	Name of the role in lowercase
Description	String (256)	Description of the role

4.4 aspnet_Users

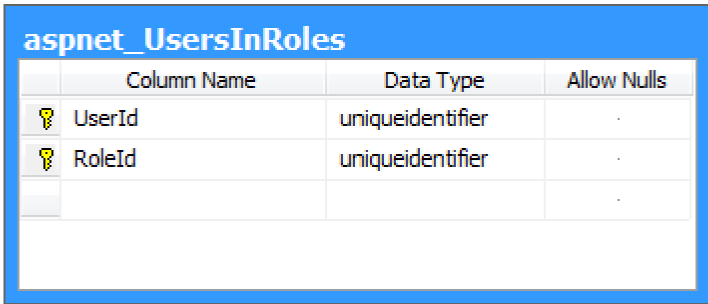
The aspnet_Users table defines each user in the system.



Field Name	Type	Description
ApplicationId	ID	Unique ID for the application
UserId (PK)	ID	Unique ID for the user
UserName	String (256)	Username
LoweredUserName	String (256)	Username in lowercase
MobileAlisa	String (16)	Mobile number
IsAnonymous	Boolean	Whether the user is anonymous
LastActivityDate	Date & Time	Last activity for the user

4.5 aspnet_UsersInRoles

The aspnet_UsersInRoles table maps users to roles.



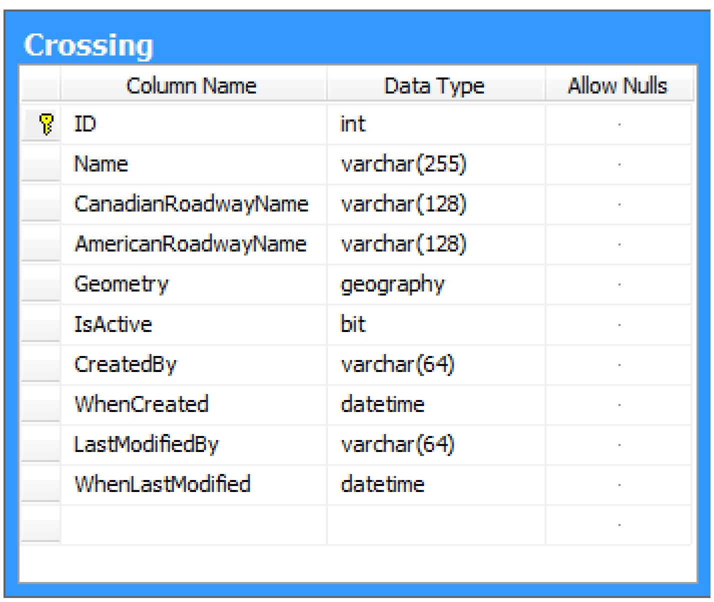
aspnet_UsersInRoles

	Column Name	Data Type	Allow Nulls
🔑	UserId	uniqueidentifier	
🔑	RoleId	uniqueidentifier	

Field Name	Type	Description
UserID (PK)	ID	Unique ID for the user
RoleID (PK)	ID	Unique ID for the role

4.6 Crossing

The Crossing Table stores the configuration of each crossing in the data warehouse. For each crossing the system stores the location as well as the timestamps for when the crossing was created and modified



Crossing

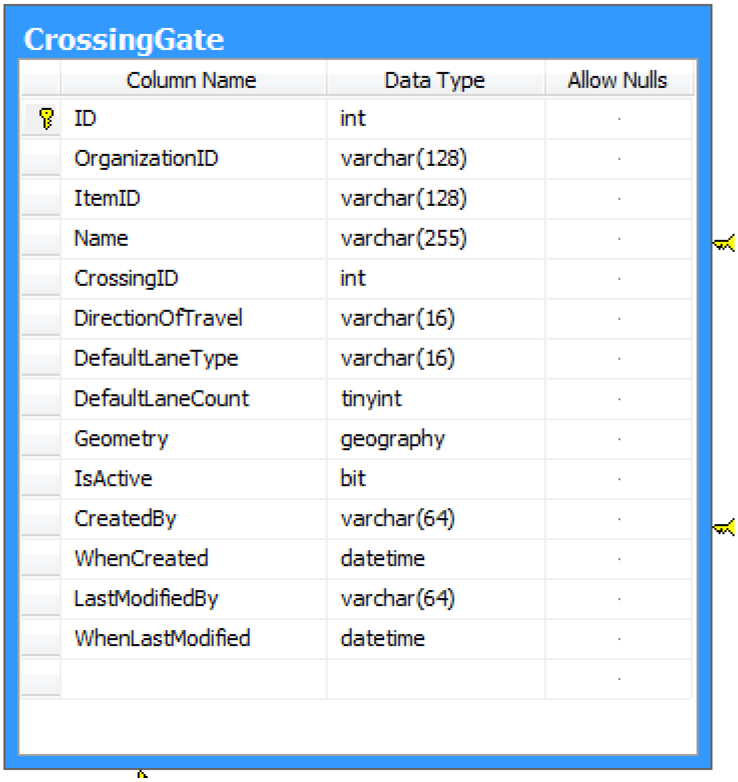
	Column Name	Data Type	Allow Nulls
🔑	ID	int	
	Name	varchar(255)	
	CanadianRoadwayName	varchar(128)	
	AmericanRoadwayName	varchar(128)	
	Geometry	geography	
	IsActive	bit	
	CreatedBy	varchar(64)	
	WhenCreated	datetime	
	LastModifiedBy	varchar(64)	
	WhenLastModified	datetime	

Field Name	Type	Description

Field Name	Type	Description
ID (PK)	Int	Id for the crossing
Name	String (255)	Display name
CanadianRoadwayName	String (128)	Name of the roadway on the Canadian side of the crossing
AmericanRoadwayName	String (128)	Name of the roadway on the American side of the crossing
Geometry	Geography	Lat/Long for the crossing
IsActive	Boolean	Whether the crossing is active
CreatedBy	String (64)	The username that created the crossing
WhenCreated	Date & Time	Date & time when the crossing was created
LastModifiedBy	String (64)	The username that modified the crossing
WhenLastModified	Date & Time	Date & time when the crossing was modified

4.7 CrossingGate

The CrossingGate Table defines the crossing lanes within the system. A crossing lane is a single lane at a crossing and direction of travel. Each lane can be associated to a particular lane type which will help with the display of the lane on the map.



CrossingGate

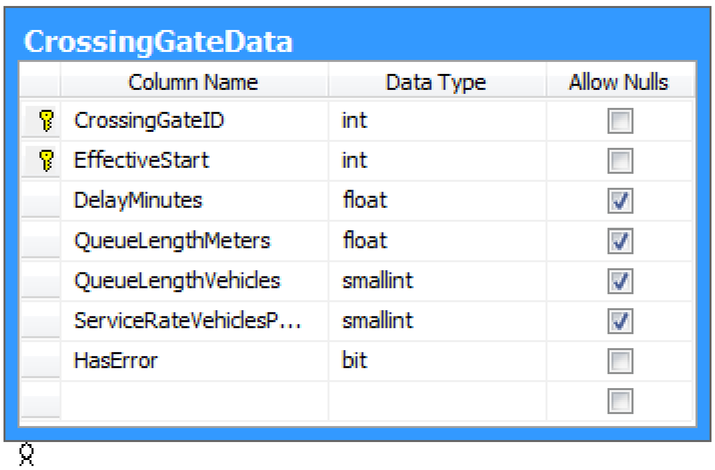
Column Name	Data Type	Allow Nulls
ID	int	
OrganizationID	varchar(128)	
ItemID	varchar(128)	
Name	varchar(255)	
CrossingID	int	
DirectionOfTravel	varchar(16)	
DefaultLaneType	varchar(16)	
DefaultLaneCount	tinyint	
Geometry	geography	
IsActive	bit	
CreatedBy	varchar(64)	
WhenCreated	datetime	
LastModifiedBy	varchar(64)	
WhenLastModified	datetime	

Field Name	Type	Description
ID (PK)	Int	Unique ID for the crossing lane
OrganizaitonID	String (128)	BCMot, WSDOT
ItemID	String (128)	The name of the crossing lane as it appears in the XML data feed
Name	String (128)	Display name for the crossing
CrossingID	Int	Reference to a crossing
DirectionOfTravel	String (16)	Northbound, Southbound, Eastbound, Westbound
DefaultLaneType	String (16)	Lane type as defined in the XML
DefaultLaneCount	Int	Number of lanes
Geometry	Geography	Lat/Long for the crossing lane
IsActive	Boolean	Whether the crossing is active
CreatedBy	String (64)	The username that created the crossing

Field Name	Type	Description
WhenCreated	Date & Time	Date & time when the crossing was created
LastModifiedBy	String (64)	The username that modified the crossing
WhenLastModified	Date & Time	Date & time when the crossing was modified

4.8 CrossingGateData

The CrossingGateData table stores the data points for a specific crossing lane. The data points include the queue length, delay and service rate. The table also stores whether the data has an error (as reported by the XML data feed). The data is indexed by CrossingGateID and EffectiveStart ensuing that the data can be quickly retrieved and also so that duplicate records for the same crossing lane and date cannot be added to the system.



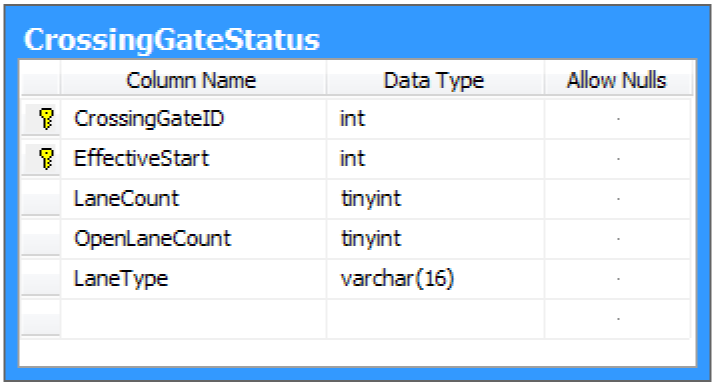
CrossingGateData

Column Name	Data Type	Allow Nulls
CrossingGateID	int	☐
EffectiveStart	int	☐
DelayMinutes	float	☑
QueueLengthMeters	float	☑
QueueLengthVehicles	smallint	☑
ServiceRateVehiclesP...	smallint	☑
HasError	bit	☐
		☐

Field Name	Type	Description
CrossingGateID (PK)	Int	Reference to the crossing lane
EffectiveStart (PK)	Int	Date & Time stored in epoch time (# of seconds from 1970)
DelayMinutes	Float	Delay in minutes as reported in the XML
QueueLengthMeters	Float	The queue length in meters as reported in the XML
QueueLengthVehicles	Int	Number of vehicles as reported in the XML
ServiceRateVehiclesPerHour	Int	Service rate as reportd in the XML
HasError	Boolean	Whether an error was reported for the data in the XML

4.9 CrossingGateStatus

The CrossingGateStatus table stores the status for each crossing lane and date & time. The status includes the number of open lanes.


CrossingGateStatus

	Column Name	Data Type	Allow Nulls
🔑	CrossingGateID	int	
🔑	EffectiveStart	int	
	LaneCount	tinyint	
	OpenLaneCount	tinyint	
	LaneType	varchar(16)	

Field Name	Type	Description
CrossingGateID (PK)	Int	Reference to the crossing lane
EffectiveStart (PK)	Int	Date & Time stored in epoch time (# of seconds from 1970)
LaneCount	Int	Number of lanes
OpenLaneCount	Int	Number of open lanes
LaneType	String (16)	Not used

4.10 CrossingGateVolume

The CrossingGateVolume table is used to store the volume for a crossing lane and date & time. The volume is stored in vehicles per hour and it continuously updated and calculated based on the mapping of detectors to crossing lanes.

CrossingGateVolume

Column Name	Data Type	Allow Nulls
CrossingGateID	int	
EffectiveStart	int	
VolumeVehiclesPerHour	float	

Field Name	Type	Description
CrossingGateID (PK)	Int	Reference to the crossing lane
EffectiveStart (PK)	Int	Date & Time stored in epoch time (# of seconds from 1970)
VolumeVehiclesPerHour	Float	Calculation of volume per hour

4.11 DelayAlertSubscription

The DelayAlertSubscription table stores the delay subscriptions. The table is used by the service in order to send out notifications to users.

DelayAlertSubscription

Column Name	Data Type	Allow Nulls
ID	uniqueidentifier	
EmailAddress	varchar(255)	
CrossingGateID	int	
AlertType	tinyint	
MinDelayMinutes	float	
IsActive	bit	
CreatedBy	varchar(64)	
WhenCreated	datetime	
LastModifiedBy	varchar(64)	
WhenLastModified	datetime	
IntervalDelayCountH...	int	
IntervalTimeHour	int	

Field Name	Type	Description
ID	Int	Unique id for the subscription
Email Address	String (255)	Email address
CrossingGateID	Int	Reference to a crossing lane
AlertType	Int	Type of alert, 0 – on change, 1 – hourly, 2 – daily
MinDelayMinutes	Int	Delay threshold
IsActive	Boolean	Whether the crossing is active
CreatedBy	String (64)	The username that created the crossing
WhenCreated	Date & Time	Date & time when the crossing was created
LastModifiedBy	String (64)	The username that modified the crossing
WhenLastModified	Date & Time	Date & time when the crossing was modified
IntervalDelayCountHour	Int	For hourly subscriptions, how many hours are the subscriptions apart
IntervalCountHour	Int	For hourly subscriptions, the start hour

4.12 Detector

The Detector table stores the configuration of the detectors in the system. Each detector can be associated to its upstream detector and can also be associated as part of a volume calculation at a crossing lane.

Detector

	Column Name	Data Type	Allow Nulls
🔑	ID	int	
	OrganizationID	varchar(128)	
	ItemID	varchar(128)	
	Name	varchar(255)	
	CrossingGateID	int	
	VolumeSortOrder	tinyint	
	UpstreamDetectorID	int	
	DefaultLaneType	varchar(16)	
	DefaultLaneNumber	tinyint	
	Geometry	geography	
	IsActive	bit	
	CreatedBy	varchar(64)	
	WhenCreated	datetime	
	LastModifiedBy	varchar(64)	
	WhenLastModified	datetime	
	WhenCrossingGateCh...	datetime	

Field Name	Type	Description
ID	Int	Unique id for the detector
OrganizationID	String (128)	BCMot, WSDOT
ItemID	String (128)	Detector ID as defined in the XML
Name	String (255)	Display name
CrossingGateID	Int	Reference to a crossing lane
VolumeSortOrder	Int	Sort order for the volume calculation
UpstreamDetectorID	Int	Reference to the upstream detector
DefaultLaneType	String (16)	Lane type for the detector (as in XML)
DefaultLaneNumber	Int	Lane number for the detector (as in XML)
Geometry	Geography	Lat/Long for the detector
IsActive	Boolean	Whether the crossing is active
CreatedBy	String (64)	The username that created the crossing
WhenCreated	Date & Time	Date & time when the crossing was created

Field Name	Type	Description
LastModifiedBy	String (64)	The username that modified the crossing
WhenLastModified	Date & Time	Date & time when the crossing was modified

4.13 DetectorCrossingLane

The DetectorCrossingLane table stores the association of Detectors to Crossing Lanes at a particular sort order.

DetectorCrossingLanes

Column Name	Data Type	Allow Nulls
DetectorID	int	
CrossingGateID	int	
VolumeSortOrder	smallint	

Field Name	Type	Description
DetectorID	Int	Detector ID
CrossingGateID	Int	Crossing Gate ID
VolumeSortOder	Int	Sort order

4.14 DetectorData

The DetectorData table is used to store the data points for detectors as received from the XML. The data is indexed by Detector ID and Effective Date Time ensuring that the data can be quickly retrieved also to prevent duplicate records for a detector at a specific date & time.

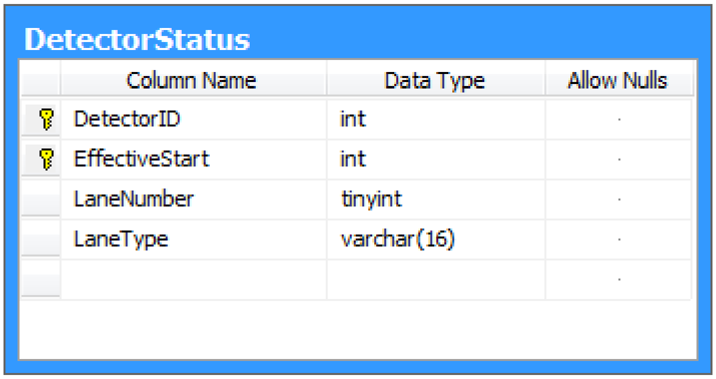
DetectorData

	Column Name	Data Type	Allow Nulls
🔑	DetectorID	int	
🔑	EffectiveStart	int	
	VolumeVehiclesPerHour	smallint	
	OccupancyPercent	tinyint	
	AverageSpeedKilomet...	tinyint	
	AverageVehicleLengt...	float	
	HasError	bit	

Field Name	Type	Description
DetectorID (PK)	Int	Detector ID
EffectiveStart (PK)	Int	Date & Time stored in epoch time (# of seconds from 1970)
VolumeVehiclesPerHour	Int	Volume in vehicles per hour (from XML)
OccupancyPercent	Int	Occupancy (from XML)
AverageSpeedKilometerPerHour	Int	Average speed in km/hr (from XML)
AverageVehicleLength	Float	Average vehicle length (from XML)
HasError	Boolean	Whether an error was reported for the data points (from XML)

4.15 DetectorStatus

The DetectorStatus table is used to store the status of each detector.



DetectorStatus

	Column Name	Data Type	Allow Nulls
🔑	DetectorID	int	
🔑	EffectiveStart	int	
	LaneNumber	tinyint	
	LaneType	varchar(16)	

Field Name	Type	Description
DetectorID (PK)	Int	Detector ID
EffectiveStart (PK)	Int	Date & Time stored in epoch time (# of seconds from 1970)
LaneNumber	Int	Lane number (XML)
LaneType	Int	Lane Type (XML)

4.16 Feed

The Feed table is used to configure the existing data feeds in the system. A data feed represents a URL that points the system at XML file location. The feed contains a file naming convention allowing the system to try to locate a file for a specific date & time. The feed can be disabled by setting the IsActive property to false.

Feed

	Column Name	Data Type	Allow Nulls
🔑	ID	int	
	UrlFormat	varchar(255)	
	ReadPeriodDays	int	
	AllowAbsentPeriodHours	int	
	IsActive	bit	
	CreatedBy	varchar(64)	
	WhenCreated	datetime	
	LastModifiedBy	varchar(64)	
	WhenLastModified	datetime	

Field Name	Type	Description
ID	Int	Unique ID for the feed.
UrlFormat	String (255)	The format of the URL, the URL, can be for both http and ftp data sources.
ReadPeriodDays	Int	Number of days that the system will look back for XML files.
AllowAbsentPeriodHours	Int	Number of hours that the system allows for absent files from the current date & time
IsActive	Boolean	Whether the crossing is active
CreatedBy	String (64)	The username that created the crossing
WhenCreated	Date & Time	Date & time when the crossing was created
LastModifiedBy	String (64)	The username that modified the crossing
WhenLastModified	Date & Time	Date & time when the crossing was modified

4.17 FeedStatus

The FeedStatus Table stores the status of each processed feed. The recorded statuses only include statuses that are outside of the AllowAbsentPeriodHours property of each data feed. The feed status table can be used in order to view which data feeds are missing.

FeedStatus

	Column Name	Data Type	Allow Nulls
🔑	FeedID	int	
🔑	EffectiveStart	int	
	Status	varchar(16)	

Field Name	Type	Description
FeedID (PK)	Int	Unique id for the feed
EffectiveStart (PK)	Int	Date & Time stored in epoch time (# of seconds from 1970)
Status	String (16)	Status for the processing of the feed. Possible values include OK, Error

4.18 GpsData

The GpsData Table is used to store GPS location data retrieved by the system. The table stores the data for particular device ids with an association location (GPS location) as well as additional info such as the Speed, Direction (heading), the mileage and a time stamp.

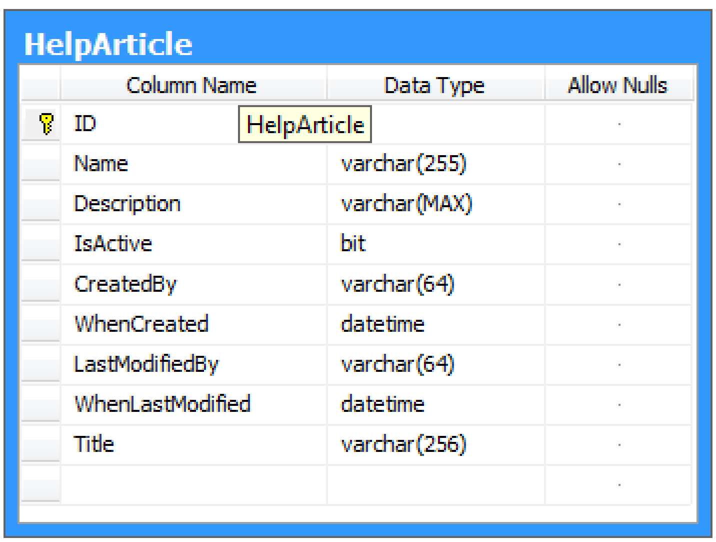
GpsData

	Column Name	Data Type	Allow Nulls
	DeviceId	varchar(32)	
	DataType	varchar(15)	
	Location	geography	
	GpsStatus	bit	
	Speed	tinyint	
	Direction	tinyint	
	Mileage	int	
	LocationTimeStamp	datetime	

Field Name	Type	Description
DeviceId	String (32)	Unique id for the device
DataType	String (15)	GPS
Location	Geography	Lat/Long for the location
GpsStatus	Boolean	On or Off
IsActive	Boolean	Whether the crossing is active
CreatedBy	String (64)	The username that created the crossing
WhenCreated	Date & Time	Date & time when the crossing was created
LastModifiedBy	String (64)	The username that modified the crossing
WhenLastModified	Date & Time	Date & time when the crossing was modified

4.19 HelpArticle

The HelpArticle table stores help content that is made available to the users throughout the website. Each help article is created in advanced and anchored to a predefined location within the website. The ability to modify the content or disable the article is available to the system administrator.


HelpArticle

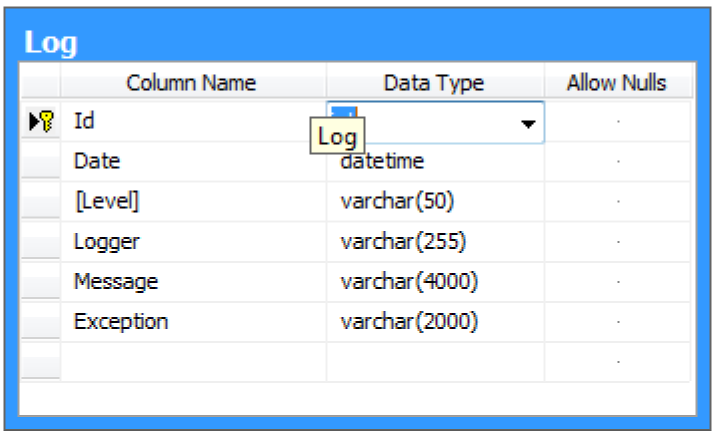
Column Name	Data Type	Allow Nulls
ID	HelpArticle	
Name	varchar(255)	
Description	varchar(MAX)	
IsActive	bit	
CreatedBy	varchar(64)	
WhenCreated	datetime	
LastModifiedBy	varchar(64)	
WhenLastModified	datetime	
Title	varchar(256)	

Field Name	Type	Description
ID	Int	Unique id for the help article
Name	String (255)	Name of the help article

Field Name	Type	Description
Description	String (max)	Actual help article. This can include HTML markup and formatting
Title	String (255)	Title of the help article
IsActive	Boolean	Whether the crossing is active
CreatedBy	String (64)	The username that created the crossing
WhenCreated	Date & Time	Date & time when the crossing was created
LastModifiedBy	String (64)	The username that modified the crossing
WhenLastModified	Date & Time	Date & time when the crossing was modified

4.20 Log

The Log Table stores the information, warning and error messages depending on the configuration of the website, database and the service components. The Log keeps track of the level (info, warning, error) as well as the message, exception and the time of the log entry.



Field Name	Type	Description
ID	Int	Unique id for the log entry
Date	Date & Time	Date & time for the log entry
Level	String (50)	Level for the log: Info, Warning, Error
Logger	String (255)	The component of the system that created the log entry

Field Name	Type	Description
Message	String (4000)	Message
Exception	String (2000)	If the level is an error additional exception details are provided.

4.21 Query

The Query Table stores the saved named reports that are available on the website. Each report is composed of numerous parameters needed to create the query.

Query

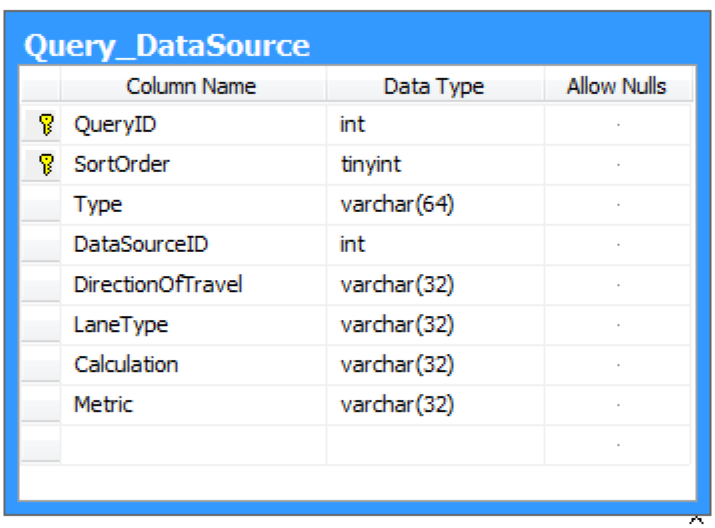
Column Name	Data Type	Allow Nulls
ID	int	
Name	varchar(255)	
GroupName	varchar(255)	
StartDate	date	
EndDate	date	
MonthsOfYear	int	
DaysOfWeek	int	
DateGrouping	varchar(32)	
TimeGrouping	varchar(32)	
CreatedBy	varchar(64)	
WhenCreated	datetime	
LastModifiedBy	varchar(64)	
WhenLastModified	datetime	

Field Name	Type	Description
ID (PK)	Int	Unique id for the query
Name	String (255)	Display name for the query
GroupName	String (255)	Grouping option for the reports. Reports with a group name will be shown together.
StartDate	Date Time	Start Date for the query

Field Name	Type	Description
EndDate	Date Time	End Date for the query
MonthsOfYear	Int	Months of the year filter stored as a bitmask (enumeration)
DaysOfWeek	Int	Days of week filter stored as a bitmask (enumeration)
DateGrouping	String (255)	Date grouping
TimeGrouping	String (255)	Time grouping
CreatedBy	String (64)	The username that created the crossing
WhenCreated	Date & Time	Date & time when the crossing was created
LastModifiedBy	String (64)	The username that modified the crossing
WhenLastModified	Date & Time	Date & time when the crossing was modified

4.22 Query_DataSource

The Query_DataSource defines the detectors and crossings that area part of a named query. Each data source defines the detector or crossing and the associated metrics.



Query_DataSource

Column Name	Data Type	Allow Nulls
QueryID	int	
SortOrder	tinyint	
Type	varchar(64)	
DataSourceID	int	
DirectionOfTravel	varchar(32)	
LaneType	varchar(32)	
Calculation	varchar(32)	
Metric	varchar(32)	

Field Name	Type	Description
QueryID (PK)	Int	Reference to a query id
SortOrder (PK)	Int	Sort order
Type	String (64)	Crossing or Detector
DataSourceID	Int	Reference to a detector or crossing id

Field Name	Type	Description
DirectionOfTravel	String (32)	Northbound, Southbound, Eastbound, Westbound
LaneType	String (32)	Lane type
Calculation	String (32)	Calculation type, ave, min, max, std, var, sum
Metric	String (32)	Metric type for either a crossing or detector: vol, spd, occ, len, etc

5 Service Description

The processing of XML files and the sending out notifications is performed by a windows service. The service is configured to continuously check for files from the configured feeds and store the data in the WCOG SQL Server database. Apart from the processing of XML, the service is also responsible for the sending out of notifications to the public subscribers and system warning emails to the system administrative staff. The details of operation are described in detail in the following sections.

5.1 Configuration

The service contains a configuration file allowing the system administrator to configure the service parameters. In order for the changes to take effect the service must be restarted. The configuration settings in the app.config file are:

Config Parameter	Description	Default Value
ConnectionString	connection string used to communicate with the WCOG SQL Server database. This string is in the standard connecting string format	Server=10.128.1.4;Database=CascadeTestMarc;User Id=sa;Password=Ibigroup10!
DefaultRepositoryUsername	default user account used by the system	System
DefaultRepositorySrid	id for the default system user	4326
MaxAllowedQuerySize	maximum number of records that can be displayed using the Custom Query	8000
EmailRecipients	comma separated list of recipients that receive the Daily Warning Email	melissa@wcog.org
EmailSubject	Subject of the Daily Warning Email	Warning Log
EnableXmlThread	Whether the xml processing should be enabled	True
EnableLogThread	Whether the logging should be enabled	True
EnableVolumeThread	Whether the volume processing should be enabled	True
EnableGpsDataThread	Whether the GPS data processing should be enabled	False – no data source provided
EnableHourlyThread	Whether the Hourly Email Notification should be enabled	True
LogEmailRunHour	Time of the Warning Email	6

IBI GROUP

DailyDelayEmailSubject	Subject of the Daily Delay Email	Daily Border Delay Report
ThresholdDelayEmailSubject	Subject of the On Change Delay Email	Current Border Delay Report
IntervalDelayEmailSubject	Subject of the hour interval delay email	Hourly Border Delay Report
GpsDataFtpUsername	Username for the FTP location for GPS data	TBD
GpsDataFtpPassword	Password for the FTP location for GPS data	TBD
GpsDataUrlFormat	URL format for the GPS location	ftp://ftp.ibigroup.com/{0:yyyyMMdd-HHmm}.csv
GpsDataFilterStartPoint	Top left Lat/Long for the bounding box that filters GPS data	="-123.99831890156247, 49.518885503202114
GpsDataFilterEndPoint	Bottom right Lat/Long for the bounding box that filters GPS data	-123.99831890156247, 49.518885503202114
SiteUrl	Site URL used to reference images and paths in the email reports	http://www.cascadegatewaydata.com
DeleteUrl	URL for the delete emails	http://www.cascadegatewaydata.com/Subscription/Delete
Feed Processor – unprocessedDirectory	The location on the computer where unprocessed XML files are stored	E:\wcogxml\unprocessed
Feed Processor – processedDirectory	The location on the computer where processed XML files are stored	E:\wcogxml\processed
Feed Processor – unusableDirectory	The location on the computer where unusable XML files are stored	E:\wcogxml\unusable

5.2 XML Data Format

The service processes email in a specific format as defined by the XML structure below. The XML schema is not available at this time, but the service looks for each of the nodes outlined below. If the service is unable to process the XML due to format or data values an exception is thrown by the service and recorded in the log. A single error in an XML file will cause none of the data in the XML file to be stored. Additional notes have been added (in red) to outline which elements are mandatory and the range of possible values. Problems with the implementation of the XML data format have been outlined in an additional document and sent to Melissa Miller for review.

<borderCrossingData xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance xsi:noNamespaceSchemaLocation="http://cascadegatewaydata.com/docs/border_data-2007-03-27.xsd">

<organization-id>WSDOT</organization-id>

<collection-period>

<collection-period-item>

<date>20110111</date> [Mandatory – date format yyyymmdd]

<start-time>116022</start-time> [Mandatory – time format HHmmss]

<end-time>120022</end-time> [Mandatory – time format HHmmss]

<measurement-duration>0</measurement-duration> [Mandatory – numeric]

<detector-reports>

```
                <detector-report>
                    <vds-id>005es27650:DON_Stn+Nx</vds-id> [Mandatory – used to reference the detector]
                    <description>Peace Arch</description>
                    <lane-data>
                        <lane-data-item>
                            <detector-id>005es27650:_MN_O_3</detector-id> [Mandatory]
                            <lane-number>1</lane-number> [Mandatory]
                            <lane-vehicle-speed>60</lane-vehicle-speed> [Mandatory – value can be null]
                            <lane-vehicle-volume>2</lane-vehicle-volume> [Mandatory – value can be null]
                            <occupancy>0</occupancy> [Mandatory – value can be null]
                            <avg-vehicle-length>65</avg-vehicle-length> [Mandatory – value can be null]
                        </lane-data-item>
                </detector-report>
            </detector-reports>
            <crossing-data>
                <crossing>
                    <crossing-id>I5</crossing-id> [Mandatory]
                    <status>Open</status> [Mandatory]
                    <lanes>8</lanes>[Mandatory]
                    <direction>n</direction>[Mandatory]
                    <delay>0</delay>[Mandatory – value can be null]
                    <open-lanes>2</open-lanes> [Mandatory – value can be null]
                    <queue-length>0</queue-length> [Mandatory – value can be null]
                    <vehicles-in-queue>0</vehicles-in-queue> [Mandatory – value can be null]
                    <service-rate>9</service-rate> [Mandatory – value can be null]
                </crossing>
            </crossing-data>
        </crossing-data>
</collection-period-item>
```

IBI GROUP

</collection-period>

</borderCrossingData>

5.3 Algorithm

The following section outlines the processing steps performed by the service. The processing is broken down by each of the various threads responsible for specific tasks within the service. On start-up of the service, a number of threads are enabled based on the thread configuration (app.config). The start time and any error are logged (db) by the service for each thread. The service monitors the threads and is responsible for keeping each thread alive. If the service stops or is stops, it performs a shutdown of each thread to ensure that a single version of each processing thread is running.

5.3.1 XML Processing Thread

The XML Processing Thread is responsible for retrieving data in the forms of XML files for each of the configured and enabled feeds, storing and committing the data to the database as well as the sending out of On Change Emails. The following steps are performed by the thread (sequence):

Get the feed configuration from the database in order to retrieve the URL format.

For each feed retrieve the full list of already processed data feeds by using the Read Period Days Parameter. This parameter tells the service how far back to look for missed feeds. Using the processed list and the Read Period Days Parameter, the service creates a list of 5 minute feeds that need to be processed based on the current date and time.

For each item in the list, the service attempts to retrieve the file based on the Feed URL. If the system cannot retrieve the file (file not available) the service records an error for the specific Feed and Date Time. Every file that is successfully retrieved (via http or ftp) is stored locally by the service in the Unprocessed Directory (configurable).

For each successfully downloaded file, the system parses the XML. This action validates the XML file and breaks the file down into Detector and Crossing data objects. Any error that occurs during the parsing is recorded by the service as Parsing Error and the file is moved to the Unusable Director (configurable).

For each successfully parsed file, the system saves the data. The service checks if the detector and crossing is known and saves the data to the following tables:

- Detector – saves detectors found in the XML file where the detector was not previously found in the database. The system administrator is notified of new detectors in the Detector Admin Screen.
- Detector Status Data – saves the status for the detector (at a time)
- Detector Data – saves the data points for the detector id and date & time, standardizing the values for each data point
- Crossing – saves crossings found in the XML file where the detector was not previously found in the database. The system administrator is notified of new detectors in the Crossing Admin Screen.
- Crossing Status Data – saves the status for the crossing (at a time)
- Crossing Data – saves the data points for the crossing id and date & time, standardizing the values for each data point.

Once the file is successfully saved and committed to the database, the service saves the file to the Processed Directory (configured)

At the end of the XML file processing the service attempts to send out threshold notifications (on change). The following steps are performed:

Service gathers information regarding the most recently processed crossing lane data.

The service checks on the Delay Threshold subscriptions stored in the database and selects subscriptions where the delay value exceeds the delay in the recently processed data.

For each subscription that matches an Email is sent out to the user in a predefined format. The notification formats are described in the Notification section below.

After performing both the processing and notification actions the thread sleeps 3 minutes and repeats the process.

5.3.2 Daily Processing Thread

The Daily Processing Thread is responsible for the sending of out of daily notifications to the public users and also warning emails to the system administrators. The thread performs these actions after midnight (configurable parameter).

The following steps are performed in the processing of system warning notifications:

For each Crossing Lane and Detector perform a data availability query for the previous date based on the current date. The resulting value represents the percentage of feeds for a crossing or a detector where data is available. The format of the email is described in the notification section

The following steps are performed in the processing of daily notification emails:

Get a list of daily subscriptions

Get a list of delays for each crossing lane and filter based on the subscriptions

Send a notification email to each subscriber. The format of the email is described in the notification secton

5.3.3 Hourly Processing Thread

The Hourly Processing Thread is responsible for sending Interval emails to the public users. Each interval subscription allows the user to specify the start hour as well as the interval and the delay that must be exceeded by the system. The following steps are performed in the processing of hourly notifications:

The thread wakes up at the hour mark

The thread retrieves all interval notifications from the database and checks if any of the subscriptions match the current hour. This check requires checking of the start dates and the interval.

For each subscription that matches the current hour, the system retrieves crossing lane delay information for the previous interval and sends the email to the subscriber. The format of the email is described in the notification section.

5.3.4 GPS Processing Thread

The GPS Processing Thread is used to retrieve and store GPS data from a configured GPS data source. The current implementation gathers data from a single configured data source and the processing of the data is performed daily (after midnight). The GPS is processed from a CSV that is made available via FTP. The format of the GPS data is in the following format (where each line in the file represents a single record):

DeviceId, Date_Type, Lattitude, Longitude, GPS_Status, Speed, Direction, Mileage, Location_Timestamp

The entire file is parsed into GPS Data objects and any parsing errors are logged to the database. The system also allows for the filtering of GPS data based on a bounding box. The bounding box is defined by a set of lat/long coordinates defined in the app.config file for the service. The filtering of GPS data helps keeping the total number of records low especially if the data files are large and contain data for a large geographic area.

5.3.5 Volume Calculation Thread

The Volume Calculation Thread is responsible for performing the calculation of volumes for crossings based on the association of detectors to crossing lanes. The calculation is performed periodically in order to make sure that the calculations are up to date. This approach was taken instead of an on demand approach since the calculations can take a number of minutes. The following steps are performed for the recalculation of volumes for any crossing lane:

> The re-calculation of volumes is performed for each crossing lane for all available crossing data. It ensures that after any detector changes all the crossing volume data is recalculated and made available to the public.
>
> The first step involves the deletion of previously calculated crossing volumes from the CrossingGateVolume table
>
> the second step is the selection of all detector data (mapped to the crossing lane) sorted by the sort order (smallest to largest). The records grouped by sort order represent lines of detectors where each sort order represents a set of detectors that can be used in the calculation.
>
> - For each group (by sort order) the system checks if all detector data for the time period is without error (hasError = 0). If there is no error the volume of the detectors is totalled and the volume for the crossing lane and time period is saved to the CrossingGateVolume table. If an error is found for any of the detectors (for a specific time period), the system will attempt to calculate the volume for the next sort order, repeating the process.

The calculation of volume for a particular crossing requires that at a minimum at least a single detector is mapped to a crossing lane. A volume of 0 will be assigned to a crossing lane in the event of no data for the mapped detectors at every sort order.

5.4 Notification

This section describes the formats of the notification emails including sample email messages

5.4.1 Daily Warning Email

The Daily Warning Email (administrators only) provides a view of the data availability for each configured crossing lane and detector. The data availability is represented as a percentage and presented graphically to be uniform with the public website. The data is sorted in ascending order providing access to the crossing lanes and detectors with low values.

Border Data Warehouse

Crossing Lane	Data Availability
Lynden/Aldergrove North Cars	66%
Pacific Highway North NEXUS	70%
Test Crossing Lane	97%
8th Avenue	99%
Peace Arch South w/8th Avenue	99%
Pacific Highway South Cars	99%
Peace Arch South Cars	99%

5.4.2 Threshold, Hourly, Daily Delay Reports

The Threshold, Hourly and Daily delay reports use the same email template. Each email provides the data in a table format and also includes a link allowing the user to unsubscribe from the email.

Border Data Warehouse

If you no longer wish to receive these emails, please click here to unsubscribe.

Daily Border Delay Data for Pacific Highway South Cars

Date/Time	Delay in Minutes
9/5/2011 12:05 AM	1
9/5/2011 12:40 AM	1
9/5/2011 12:45 AM	1
9/5/2011 12:50 AM	1

Appendix H
XML Issues Report

Whatcom Council of Governments

CASCADE GATEWAY XML PROBLEMS

REPORT

SEP 26, 2011

IBI GROUP

IBI GROUP

1 Introduction

The following document outlines the problems in the xml files processed for the Cascade Data Warehouse (www.cascadegatewaydata.org). The site is used to collect data from 2 XML data sources that both adhere to a common XML data format.

2 XML Data Format

The data format used by the data warehouse is presented in the section below. It is presented as an xml document and not a schema.

```
<borderCrossingData xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:noNamespaceSchemaLocation="http://cascadegatewaydata.com/docs/border_data-2007-03-27.xsd">
    <organization-id>WSDOT</organization-id>
    <collection-period>
    <collection-period-item>
        <date>20110111</date>
        <start-time>116022</start-time>
        <end-time>120022</end-time>
        <measurement-duration>0</measurement-duration>
        <detector-reports>
            <detector-report>
                <vds-id>005es27650:DON_Stn+Nx</vds-id>
                <description>Peace Arch</description>
                <lane-data>
                    <lane-data-item>
                        <detector-id>005es27650:_MN_O_3</detector-id>
                        <lane-number>1</lane-number>
                        <lane-vehicle-speed>60</lane-vehicle-speed>
                        <lane-vehicle-volume>2</lane-vehicle-volume>
                        <occupancy>0</occupancy>
                        <avg-vehicle-length>65</avg-vehicle-length>
                    </lane-data-item>
            </detector-report>
        </detector-reports>
        <crossing-data>
```

<crossing>
<crossing-id>I5</crossing-id>
<status>Open</status>
<lanes>8</lanes>
<direction>n</direction>
<delay>0</delay>
<open-lanes>2</open-lanes>
<queue-length>0</queue-length>
<vehicles-in-queue>0</vehicles-in-queue>
<service-rate>9</service-rate>
</crossing>
</crossing-data>
</crossing-data>
</collection-period-item>
</collection-period>
</borderCrossingData>

The following describes each of the nodes and any associated data problems (in red)

Date – date for the xml file. This date is in format yyyymmdd.

Start Time / End Time – start & end time for the data. The problem with these two nodes is that the data very frequently is not 5 minutes apart and in some cases it does not create a proper date. Dates such as 11:60:22 are found. These two fields cannot be accurately used so the Cascade Gateway Data Warehouse depends on the filename for the date and the start time.

Measurement Duration – the measurement duration should correspond to interval for the data and should be 300 seconds.

Crossing Data & Detector Report – these nodes generally contain proper values, but in some cases have produced -1 for volume and speed. It is assumed that -1 are invalid data values and a suggestion is to mark the report for the detector or crossing as invalid.

3 Empty XML Files

Another potential issue are files are empty XML files (0 kb). It is unclear what causes this, but if the cause is that no data was received during the 5 minute collection period the file should still contain the header information and have empty detector reports and crossing data nodes. This problem occurs from both data feeds.

IBI GROUP

4 Incomplete XML Files

Some of the XML is also incomplete, looking like it was either cut-off or not formed properly. An example of this is the following (entire XML file contents). This type of error occurs from the BCMOT data feed.

<date>20101229</date>

<start-time>154000</start-time>

<end-time>154500</end-time>

<measurement-duration>300</measurement-duration>

<crossing-data>

</crossing-data>

<detector-reports>

</detector-reports>

Appendix I
Duplicate Records Report

Whatcom Council of Governments

DUPLICATE RECORDS

REPORT

SEP 26, 2011

IBI GROUP

1. INTRODUCTION

The following document outlines the duplicate records that existed in the first version of the Cascade Gateway Data Warehouse.

2. DUPLICATE RECORDS

The first version of the Cascade Gateway Data Warehouse stored crossing and detector data in two tables, CrossingFacts and DetectorFacts. These tables suffered from a number of problems in structure including lack of indexes and constraints allowing duplicate records to be found. Both of these problems have been addressed in the new version resulting in improved performance (speed of data retrieval) as well as securing the data from duplicates.

2.1 Crossings

The crossing data contained about 5 million records with the following duplicates detected based on the Crossing ID, Date and Time. These 3 keys should uniquely identify a record and duplicates should not exist.

> 55,954 – duplicate records based on the 3 keys
>
> 127,099 – total number of duplicate records found (this shows that some crossing records were inserted multiple times

The duplicates were caused by the lack of indexes and constraints on the table preventing a record identified by the 3 keys (ID, Date, Time) to be inserted more than once. The primary cause for the duplicates was the re-parsing of some live data files and not problems with the actual data in the XML feeds. This problem has been addressed in the new version with the introduction of unique keys and constraints on the Crossing and Detector tables.

2.2 Detectors

The Detector data contained about 84 million records with duplicates detected based on the Detector ID, Date and Time.

> 1,520,906 - duplicates based on the 3 keys
>
> 3,266,683 – total number of duplicates in the system (573,865 without a detector id or timestamp)

The problems with detector data were the same as with the crossings, lack of indexes and unique constraints preventing duplicates from being entered into the system. As with the crossing data the duplicates were created due to parsing of the same XML file multiple times and the system not preventing the data from being added to the db.

Appendix J
Integration of Additional Data Sources (Phase II)

Whatcom Council of Governments

BORDER DATA WAREHOUSE UPGRADE
INTEGRATION OF ADDITIONAL DATA SOURCES

TECHNICAL MEMORANDUM

SEP 26, 2011

IBI GROUP

TABLE OF CONTENTS

1. INTRODUCTION

The purpose of this Technical Memorandum is to develop a preliminary design scope for the integration of additional data into the newly upgraded Border Data Warehouse (BDW).

1.1 Background & Objectives

The expanded BDW was envisioned to include access to other important border related data in the Cascade Gateway. These were not included in the original scope of the BDW upgrade project because the availability of the data was uncertain at the time, relative to the schedule and urgency of completing the upgrade project itself.

The data under consideration includes:

Booth status data to improve northbound and southbound ATIS systems

WSDOT Weigh-in-Motion detector data; this WIM station is just south of I-5 Exit 275.

BCMOT Weigh-in-Motion detectors; BCMOT owns three detectors (Hwy 15, 13, and 11).

CVISN data: Several detectors along I-5 between the Port of Seattle and the border are available for acquiring data.

GPS fleet data: TRAC and WSDOT have purchased several years of commercial fleet GPS location data from several vendors and may be able to make some of that data available for display in the archive.

BTS data feed: The BTS trans-border surface freight database is an online query tool that is regularly used by IMTC partners and would be useful to have accessible within the BDW

1.2 Document Format

This document describes each of the data sources that was considered during the data source review and includes:

BTS data

CVSN data

GPS data

WS DOT WIM data

For each data type the available data types are described including the data format, how the data can be obtained and any recommendations for how the data should be used.

2. BTS DATA FEED

2.1 Available Data

The current BTS data includes commodities by either value (dollars), weight. The data is aggregated for each commodity by year/month. The following table describes the data available:

Data	Description	Data Type
STATYR	Statistical Year	Numeric – 4 digits
STATMO	Statistical Month	Numeric – 2 digits
DEPE	District and Port of Entry and Exit	Numeric – 4 digits
COMMODITY	2 digit commodity code, full list of commodity descriptions is provided	Numeric - 2 digits
TTV_T	Total Trade Value by Truck (in U.S. dollars)	Numeric
IMPV_T	Imports Value by Truck (in U.S. dollars)	Numeric
EXPV_T	Exports Value by Truck (in U.S. dollars)	Numeric
TTV_R	Total Trade Value by Rail (in U.S. dollars)	Numeric
IMPV_R	Imports Value by Rail (in U.S. dollars)	Numeric
EXPV_R	Exports Value by Rail (in U.S. dollars)	Numeric
TTWkg_T	Total Trade Weight by Truck (in Kilograms)	Numeric
TTWlb_T	Total Trade Weight by Truck (in Pounds)	Numeric
IMPWkg_T	Imports Weight by Truck (in Kilograms)	Numeric
IMPWlb_T	Imports Weight by Truck (in Pounds)	Numeric
EXPWkg_T	Exports Weight by Truck (in Kilograms)	Numeric
EXPWlb_T	Exports Weight by Truck (in Pounds)	Numeric
TTWkg_R	Total Trade Weight by Rail (in Kilograms)	Numeric
TTWlb_R	Total Trade Weight by Rail (in Pounds)	Numeric
IMPWlb_R	Imports Weight by Rail (in Pounds)	Numeric
EXPWkg_R	Exports Weight by Rail (in Pounds)	Numeric

EXPWlb_R	Exports Weight by Rail (in Pounds)	Numeric

The data can be retrieved from the database using a variety of query (filter) options. The filter options can be used to limit the time scope, geographic location as well as the commodities. The following table describes the available filter options:

Filter Name	Filter Description
Custom Ports	All Ports or any number of specific port names
Trading Partner	NAFTA (both Canada and Mexico Canada Mexico
Measure	Value or Weight
Mode	All, or any number of specific modes
Year	
Month	Annual Summary, All or specific months.
Commodity	All or any number of specific commodities
Trade Type	Total Trade or Exports or Imports
Data Display	Actual, In Millions, In Billions
Output Option	With or without % change
Other Options	Various options for the output data and handling of N/A

2.2 Options for Accessing Data

The BTS data is available via a web page accessible via: http://www.bts.gov/programs/international/transborder/TBDR_QAPC07.html. The site currently allows the user to query the site by the filter options described in the table above. The data is made available to the user in CSV file.

In order to make the BTS data available through the WCOG BDW, the user will be able to make a query by selecting the relevant options in the BTS Custom Query screen, or by using the BTS API. Both options will create a query based on the URL parameters made available by the BTS website and retrieve the data from the BTS system in real-time. The retrieved data will not be stored in the data warehouse and the WCOG data warehouse will act as a front end for the BTS data. The advantage of not storing the data is to prevent data duplication and also due to the complexity of keeping both data sets synchronized. The retrieved data will be displayed to the end user using the same techniques and formats as all other data and include tables, CSV, JSON and also the Visualization Charts.

2.3 Recommended Design / Approach

IBI recommends the following approach to be used to incorporate the BTS data in the warehouse. The following sections outline the tasks to be performed as well as the time estimates.

1. **Design** – to include the design of the BTS Query Input and making decisions regarding what options should be made available to the user. The Outcome of the design is a screen mock-up.

2. **Data Analysis** – study the details of the data format as well as the current query capabilities of the BTS system. The outcome of the analysis is the data structure for the query and output data.

3. **Query Input Screen** – Creation of the BTS query input screen based on the mock-up of the design.

4. **Table Output** – Create the BTS table output page based on the current implementation.

5. **BTS Chart Output** – Create the BTS chart page based on the current implementation.

6. **BTS CSV Output** – Create the CSV Output based on the current implementation.

7. **BTS JSON Output** – Create the JSON output based on the current implementation.

8. **Save Query** – This option allows the administrator to save any BTS queries and make them available to the public.

9. **Testing** – internal testing

10. **Deployment** – deployment to the live environment

2.4 Schedule & Pricing

We estimate that this work can be completed over the available three week period this June, allocating one week to detailed design, one to development, and one to testing and commissioning. We estimate that this work will require 3 person weeks of effort, which using a blended rate of $175 per hour (and a total of 112 hours) translates to $20,000.

3. CVISN DATA

3.1 Available Data

The CVISN data includes count data from transponder equipped trucks that is collected at reader locations. There are a number of readers in the border location and data will be collected from these readers in order to estimate the delay and travel times at the border.

The following table describes the data available (based on the description of the data feed, we have not received the actual data feed at this point)

Data	Description	Data Type
Reader ID	Either ID or Description	Numeric
Truck ID	Unique identifier for the truck	Numeric
Date/Time	Date/Time for the read	Date

The exact data format for the data has not been established but will be confirmed as part of our detailed design activities.

3.2 Options for Accessing Data

The options for how the data can be accessed have not been established, but based on the discussion the data will be available via FTP.

3.3 Recommended Design / Approach

IBI suggests the following approach for collecting and calculating delay data using the CVISN data:

Collect data from all reader locations using FTP

Configure reader locations into pairs (one on the Canadian side and one on the American side). Each pair has a specific distance between the 2 readers and this distance will be used in the calculation of the speed the truck travelled between the 2 readers.

Match truck trips (north to south and south to north) based on the Truck Id. For each truck trip calculate the amount of time it took for the truck to travel the distance between the 2 readers

Configure a free flow speed for each reader pair.

Calculate the “Delay” for the crossing based on the time and free flow speed.

Problems & Issues

The calculated delay treats the amount of time at the border by each truck the same way, meaning that trucks that get inspected cannot be differentiated from trucks that made it through quickly. Statistical approaches would have to be used.

The following sections outline the tasks to be performed as well as the time estimates.

1. **Design** – to include the analysis of the data formats and how the data should be saved into the WCOG data warehouse.

2. **Data Analysis** – to include the analysis of the collected data and its viability for delay calculations

3. **Data Collection** – creation of the needed database structures and changes to the data collection services to begin collection.

3.4 Schedule & Pricing

We estimate that this work can be completed over the available three week period this June, allocating one week to detailed design, one to development, and one to testing and commissioning. We estimate that this work will require 3 person weeks of effort, which using a blended rate of $175 per hour (and a total of 112 hours) translates to $20,000.

4. GPS FLEET DATA

4.1 Available Data

The GPS data is collected from GPS system vendors from approximately 6000 equipped trucks. The following data is available for each truck:

Data	Description	Data Type
Truck Id	Unique identifier for a truck	Numeric
DateTime	Date & Time for the read	Date & Time
Heading	Direction of travel	String
Latitude		Numeric
Longitude		Numeric
Vehicle Status	Whether the vehicle is active	String

4.2 Options for Accessing Data

The data is available for the trucks via FTP and truck locations are updated every 5-10 minutes.

4.3 Recommended Design / Approach

IBI proposes to begin collecting of GPS data in the border region. The data will be retrieved from the FTP and filtered using pre-defined collection geo-fences. The collection geo-fences will be sufficiently large in order to perform calculations about the volume and delay at a specific border crossing.

Using the GPS data a number of approaches can be used to calculate the delay (and possibly volume) at a specific crossing. The options are outlined below:

Truck Trips – create truck trips from the data in the border region. Each truck trip is assessed against a geo-fence at the border and the delay is calculated as the time when the truck enters and leaves the geo-fence (after crossing the border)

Truck Density – an alternate approach is to estimate delay based on the density of truck reads within a geo-fence. The thought is that the more reads that are currently within the geo-fence at the border the higher the delay.

Problems & Issues

The calculation of delay is difficult with either approach without the ability to study the data and its granularity in detail.

The GPS data is not very useful when the delay is low due to the 5-10 minute GPS read delay.

The following sections outline the tasks to be performed as well as the time estimates.

1. **Design** – to include the analysis of the data formats and how the data should be saved into the WCOG data warehouse.

2. **Data Analysis** – to include the analysis of the collected data and its viability for delay calculations

3. **Data Collection** – creation of the needed database structures and changes to the data collection services to begin collection.

4.4 Schedule & Pricing

We estimate that this work can be completed over the available three week period this June, allocating one week to detailed design, one to development, and one to testing and commissioning. We estimate that this work will require 3 person weeks of effort, which using a blended rate of $175 per hour (and a total of 112 hours) translates to $20,000.

5. WSDOT WEIGH-IN-MOTION DETECTOR DATA

The integration of the WSDOT WIM data into the BDW is under consideration at this time; it may be feasible to complete this integration as part of the scope under consideration for the end of June 2011. We anticipate that the budget for this integration will be similar to the other data elements presented in the previous sections; we recommend that a contingency budget of $20,000 be reserved for inclusion of this data into the BDW; the use of this contingency budget would be approved by WCOG provided that supporting design information is provided by IBI Group, with a commitment that the work can be completed by the end of June 2011.

Appendix K
Integration of Additional Data Sources (Phase III)

Whatcom Council of Governments

BORDER DATA WAREHOUSE UPGRADE
INTEGRATION OF ADDITIONAL DATA SOURCES (PHASE 3)

REPORT

SEP 26, 2011

IBI GROUP

1 Introduction

The following document describes the work performed by IBI Group as part of the Phase 3 development of the Cascade Data Warehouse. The work included the integration of the BTS Data Feed as well as the integration of the GPS Fleet data.

The BTS data integration included the creation of query screens for BTS data inside the Cascade Data Warehouse and outputting of the query. The actual query and the data is stored by BTS and the data is queried on request (as needed)

2 BTS Data Feed

The BTS Freight Data screen allows the public users to query the BTS Freight Data System. The cascade data warehouse acts as a query front end which dispatches the calls to the system and displays the information in a manner uniform with the rest of the site.

The BTS Freight query allows the user to select the following input parameters:

Year – year for the query

Measure – Value or Weight

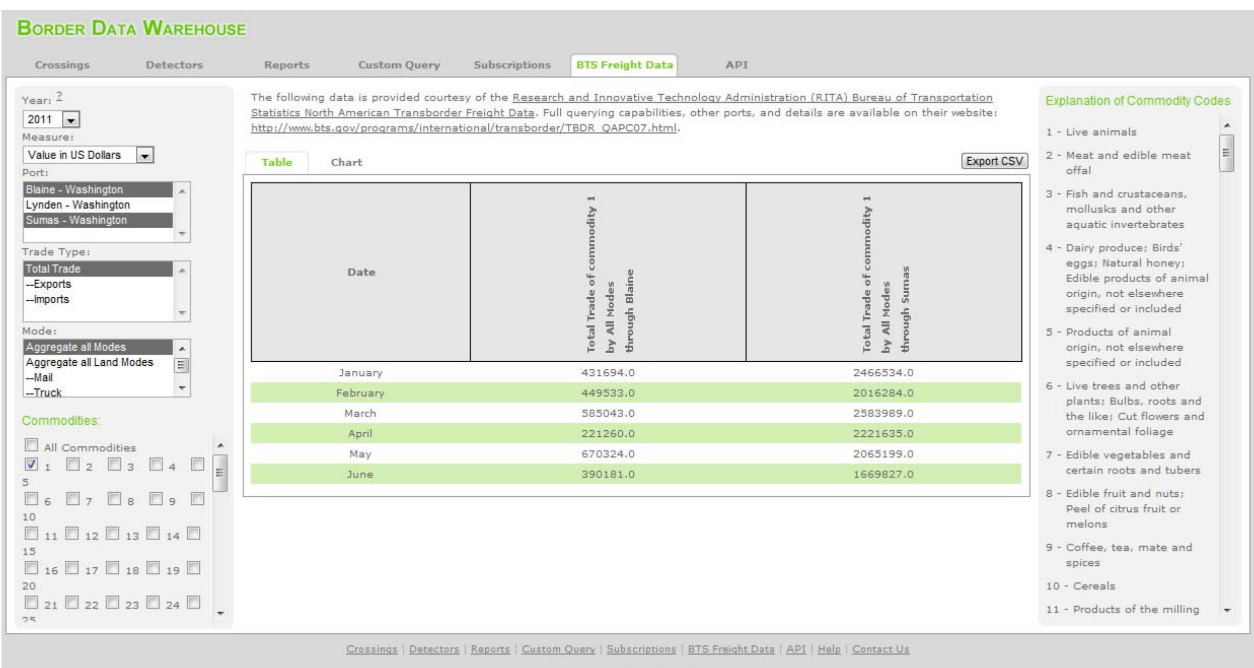
Port – allow the user to select multiple values from Blaine, Lynden and Sumas

Trade Type – Both, Export, Import

Mode – allow the user to select from any of the predefined land, air and sea modes

Commodities – allow the user to select any number of commodity codes.

The query that is performed by the BTS Freight Data System is displayed using the standard Table, Chart and Export Views providing a consistent data output across the entire site.



BTS Freight Query

The BTS query capability follows the architecture of the Cascade Gateway Data Warehouse and this architecture is described fully in he WCOG Design Document.

2.1 Recommendations & Enhancements

The following list represents possible recommendations and enhancements to the BTS Data Query functionality:

- Ability to total the values in the table
- Formatting of the data with $ and 1000 separators (,)

3 GPS Fleet Data

The Cascade Gateway Data Warehouse has been upgraded to store GPS Fleet Data information. The system has been setup to accept GPS data via FTP, parse the data from a predefined data format and save the data to the database.

The available GPS data gives the following information for each truck:

Data	Description	Data Type
Truck Id	Unique identifier for a truck	Numeric
DateTime	Date & Time for the read	Date & Time
Heading	Direction of travel	String
Latitude		Numeric
Longitude		Numeric
Vehicle Status	Whether the vehicle is active	String

The data presented above gives a unique Truck ID, Timestamp as well as the location of the truck at that time specified by a Lat/Long. The data is contained in large data files that are retrieved from a configurable FTP location (potentially with username / password). The data files must be in the following CSV format with a header row where the field names match the description above.

DEVICE_ID,DATA_TYPE,LATITUDE,LONGITUDE,GPS_STATUS,SPEED,DIRECTION,MILEAGE,LOCATION_TIMESTAMP

3.1 Data Processing

The GPS Processing Thread is used to retrieve and store GPS data from a configured GPS data source. The current implementation gathers data from a single configured data source and the processing of the data is performed daily (after midnight). The GPS is processed from a CSV that is made available via FTP. The format of the GPS data is in the following format (where each line in the file represents a single record):

1. DeviceId, Date_Type, Lattitude, Longitude, GPS_Status, Speed, Direction, Mileage, Location_Timestamp

The entire file is parsed into GPS Data objects and any parsing errors are logged to the database. The system also allows for the filtering of GPS data based on a bounding box. The bounding box is defined by a set of lat/long coordinates defined in the app.config file for the service. The filtering of GPS data helps keeping the total number of records low especially if the data files are large and contain data for a large geographic area.

3.2 GPSData Table

The GpsData Table is used to store GPS location data retrieved by the system. The table stores the data for particular device ids with an association location (GPS location) as well as additional info such as the Speed, Direction (heading), the mileage and a time stamp.

GpsData

Column Name	Data Type	Allow Nulls
DeviceId	varchar(32)	
DataType	varchar(15)	
Location	geography	
GpsStatus	bit	
Speed	tinyint	
Direction	tinyint	
Mileage	int	
LocationTimeStamp	datetime	

Field Name	Type	Description
DeviceId	String (32)	Unique id for the device
DataType	String (15)	GPS
Location	Geography	Lat/Long for the location
GpsStatus	Boolean	On or Off
IsActive	Boolean	Whether the crossing is active
CreatedBy	String (64)	The username that created the crossing
WhenCreated	Date & Time	Date & time when the crossing was created
LastModifiedBy	String (64)	The username that modified the crossing
WhenLastModified	Date & Time	Date & time when the crossing was modified

3.3 Recommendations & Enhancements

The following list represents possible recommendations and enhancements to the BTS Data Query functionality:

- Collection of data from a known GPS source. This can be done in an ongoing basis or to parse and save a known dataset (1 year worth of data)
- Representation of the data on a google map. The map should include the capability to view the data as a snapshot (plotted points) for a selected timeframe and truck (s) as well as the capability to "play" the data for a specific vehicle. The play feature will allow the user to select the speed and keep the location of the vehicle in the map view
- GPS Fleet data border delay calculations. Implement algorithms that estimate the delay at the border based on live GPS fleet data. This approach would need to implement a known algorithm.
- Origin / Destination reports. Outline the frequently travelled routes based on the data.