License Mobile

Author: Victoria Castillo, December 2018

# Overview

The License Mobile application is used to look up individual and business licenses issued by the City of Vaughan. The application is owned by By-Law and Compliance, Licensing and Permit Services.

Dev: <http://v-apphostdev/LicenseMobile>  
Prod: <http://v-apphostsrv/LicenseMobile>

The personal licenses portion retrieves data from the AMANDA database through Web services in the ASP.NET MVC framework.

The business licenses portion retrieves data from the AMANDA database through Web services in the ASP.NET Web forms framework.

The application also uses scripts from Bootstrap v3.3.2 and jQuery 1.9.1

# Changelog

**Published build on apphostdev:** March 5, 2019: 6099: Personal: Resolved missing car info if org name is empty

Feb 27, 2019: 6097: Personal: Fixed mixed up make and model year outputs, resolved city license search if no non-numeric separator in search query

Feb 25, 2019: 6096: Personal: modified behavior of search criteria priority, per stakeholder’s request. Minor code comments to indicate deprecated class properties.

Feb 21, 2019: 6095: Personal: Removed try-catch from foreach as it resulted in skipped records, resolved instead with conditional statements to check whether data returned by getFolderInfoByInfoCode (car data and operating name) is good; Search criteria will only pass values with populated text input (previously passing empty values if no user input, also resulting in skipped records); Revised priority order of individual queries (city license, VIN, etc.)

Feb 7, 2019: 6082: New BR: Added city license and organization name search and fields, operating name field

Jan 25, 2019: 6073: New dev web service URLs, added logic to retrieve city and postal code from address line 2 if those columns are empty or null, exposed license type search for dev testing

Jan 22, 2019 – 6063: \*\*prod config\*\* Personal: Fixed issue of skipped records due to DOB error; Populating with addressline2 data if city/postal are empty. Both: Sorted results by descending folderRSN and expiry date

Jan 21, 2019 – 6061: Note: this is the production configuration, as the dev web service and database have been taken down.

6060: Web.config clean-up: removal of connection strings and extraneous references from previous versions (NHibernate, Entity Framework, Oracle)

Jan 17, 2019 – 6053: Build with production Web services in HTTPS but dev database. PersonModel.cs and SearchData.cs updated for search term string trimming and corrected license number check due to errors.

Jan 11, 2019 – 6046: Fixed issue with mismatched car info

Jan 10, 2019 – 6045: Personal: notification if over 10 results found, exception handling reinstated

Jan 8, 2019 – 6038: Personal photo display, exception handling, clean-up of old unneeded references, classes, methods

Jan 2-4, 2019—6034, 6035, 6036: Birthdate, ongoing corrections per comments from Hongli and Anthony (hide licence type. Navbar fixes)

Dec 20, 2018: 6030: Personal Licenses using Web services

Dec 17, 2018 – 6024: Consolidation of business license and personal license updates

Dec 13, 2018 – 6016: Implemented Web services for business licenses

Nov 1, 2018 - 5995: Updated personal licenses to new Oracle DB

Sept, 2015 – 4952: Updated to Episuite

# Web Services

Dev: <http://intranet/axis/>

Security Service: http:// intranet /axis/services/WSAmandaSecurityService?wsdl

Amanda Service: [http:/ intranet /axis/services/WSAmandaService?wsdl](http://dev-amndweb01:9080/axis/services/WSAmandaService?wsdl)

Prod: [https:// intranet /axis/](https://srv-amandamob01:8443/axis/)

Security Service: [https:// intranet /axis/services/WSAmandaSecurityService?wsdl](https://srv-amandamob01:8443/axis/services/WSAmandaSecurityService?wsdl)

Amanda Service: [https:// intranet /axis/services/WSAmandaService?wsdl](https://srv-amandamob01:8443/axis/services/WSAmandaService?wsdl)

## Running Web Services on SOAP UI

You can install SOAP UI to explore the Web services. After installing, here are the basic steps to get started:

1. Create a new SOAP project and add the dev security and service WSDLs
2. Run the authenticatePublicUser method from the security WSDL to get the lid:
   1. Username: [xxxxxxx](mailto:publicsearch@vaughan.ca), Password: xxxxxx
   2. In the bottom left corner of the request window, add the values: header: system, value: xxxxx
3. Run your service methods with the header “lid” and the lid value from the authenticatePublicUser method (as highlighted in the screencap above)

For full descriptions, signatures, and definitions of the classes and methods of the Web services, which will also be mentioned in the rest of this document, refer to the Web services’ documentation from CSDC:

Documentation: O:\6.1.8.0 AMANDA Web Services.pdf

# Deploying to Prod

1. Update from dev to prod Web Services
   1. In the Solution Explorer, under Connected Services, right-click on each WS and select Configure Service Reference…
   2. Replace the URLs (see above) to regenerate the WS methods
2. Update the client code in AmandaWS, PersonModel, LicenceApp, and Property2015 to the HTTPS endpoint names for the security and Amanda Web services.  
   **Important:** Do not “Find All” and “Replace All”—only replace the instances in the 3 CS files mentioned above. Replacing all may mess up the generated WS code.
   1. Find 3 instances in PersonModel, Property2015, and LicenceApp: WSAmandaServiceHttpSoap11Endpoint
   2. Replace: WSAmandaServiceHttpsSoap11Endpoint
   3. Only in AmandaWS.cs: WSAmandaSecurityServiceHttpSoap11Endpoint
   4. Replace: WSAmandaSecurityServiceHttpsSoap11Endpoint
3. Check the Web.config file because it gets regenerated when you change the WS reference URLs:
   1. Add the port number (8443) in the endpoint addresses in the Web.config file (because of HTTPS)
   2. If necessary, increase the maxReceivedMessage to 2000000 in the binding (see below for snippet/example)—not doing this will result in errors for the Business License search
4. Update the connection cache name (system header) in the AmandaWS.cs model class file from dev (amnddv) to prod (amndprod)
5. Build and publish the production package per usual
6. After the successful ICC, revert back to dev configuration when checking into TFS, and republish on apphostdev (if necessary).

## Web.config snippet for Prod WS

<system.serviceModel>

<bindings>

<basicHttpBinding>

<binding name="WSAmandaServiceSoap11Binding" maxReceivedMessageSize="2000000">

<security mode="Transport" />

</binding>

<binding name="WSAmandaServiceSoap11Binding1" maxReceivedMessageSize="2000000" />

<binding name="WSAmandaSecurityServiceSoap11Binding">

<security mode="Transport" />

</binding>

<binding name="WSAmandaSecurityServiceSoap11Binding1" />

</basicHttpBinding>

<customBinding>

<binding name="WSAmandaServiceSoap12Binding">

<textMessageEncoding messageVersion="Soap12" />

<httpsTransport />

</binding>

<binding name="WSAmandaSecurityServiceSoap12Binding">

<textMessageEncoding messageVersion="Soap12" />

<httpsTransport />

</binding>

</customBinding>

</bindings>

<client>

<endpoint address="https://srv-amandamob01:8443/axis/services/WSAmandaService.WSAmandaServiceHttpsSoap11Endpoint/"

binding="basicHttpBinding" bindingConfiguration="WSAmandaServiceSoap11Binding"

contract="WSAmandaService.WSAmandaServicePortType" name="WSAmandaServiceHttpsSoap11Endpoint" />

<endpoint address="https://srv-amandamob01:8443/axis/services/WSAmandaService.WSAmandaServiceHttpsSoap12Endpoint/"

binding="customBinding" bindingConfiguration="WSAmandaServiceSoap12Binding"

contract="WSAmandaService.WSAmandaServicePortType" name="WSAmandaServiceHttpsSoap12Endpoint" />

<endpoint address="https://srv-amandamob01:8443/axis/services/WSAmandaSecurityService.WSAmandaSecurityServiceHttpsSoap11Endpoint/"

binding="basicHttpBinding" bindingConfiguration="WSAmandaSecurityServiceSoap11Binding"

contract="WSAmandaSecurity.WSAmandaSecurityServicePortType" name="WSAmandaSecurityServiceHttpsSoap11Endpoint" />

<endpoint address="https://srv-amandamob01:8443/axis/services/WSAmandaSecurityService.WSAmandaSecurityServiceHttpsSoap12Endpoint/"

binding="customBinding" bindingConfiguration="WSAmandaSecurityServiceSoap12Binding"

contract="WSAmandaSecurity.WSAmandaSecurityServicePortType" name="WSAmandaSecurityServiceHttpsSoap12Endpoint" />

</client>

</system.serviceModel>

# Personal License

## Overview

The personal license search allows users to look up people’s licenses in relation to business operations. It searches Amanda’s People, Folder, and InfoCode tables.

The search is configured to return a maximum of the top 10 personal license matches even if more matches are found in the database, and the user is warned to narrow the search field.

## Models

AmandaWS.cs

protected static string GetTokenLid()

This class instantiates an Amanda Security Service port client on a SOAP 1.1 endpoint and returns a token string from the Web service, which is required to pass into HTTP headers to consume the Web services.

The following lines of code contain values that need to be changed for production:

requestProperty.Headers["system"] = "xxxxxdev";

requestProperty.Headers["system"] = "xxxxxxprod";

* The name of the connection cache to use. Dev and production respectively (obviously).

auth\_token = asec.authenticateUser("username", "password");

* The user ID and password, respectively, to connect to the database. (Note: these credentials also work for the production database, so no need to change it)

public static string Lid

Insert this string into an HTTP header of an Amanda Service port client to call on the Web service methods.

Connects to the Amanda Security Web service with the login credentials

BlobImage.cs

- Deprecated upon migration to Web service (NHibernate)

Card.cs

- Contains license details, populated by data from the Web service

DisplayInformation.cs

- Contains an object of Person and an object of Card, and an int of imageCount (retired)

LicenceTypes.cs

- Calls on a custom SQL transaction Web service method to populate a drop-down list of license types in the Finder

PageInfo.cs

- Data to support pagination

Person.cs

- Person’s individual data

PersonModel.cs

public List<DisplayInfomation> displayInfoList { set; get; }

A list of DisplayInformation objects, which consists of an object each of Person and Card, containing the search results, that gets passed to the controller

public SearchData searchData { set; get; }

An object of the search terms entered by the user

public bool OverTen { set; get; }

Boolean value if over ten search results are found, flag for a warning message on the view that is passed through the controller

public void ExecuteQuery()

The AMANDA Web service is called to search for license records using the entered search terms (SearchData object), and a list of DisplayInformation objects is passed to the GuardCardController.

## Web Services

### Search Criteria

Refer to p. 97, Illustration: Search Web-Services portion of the Amanda Web Services documentation for the description.

**Note**: To assign to the operator, negative, and cojuctiveOperator members, you should also set operatorSpecified, negativeSpecified, and conjuctiveOperatorSpecified Boolean values to true. This drove me crazy because this was never explained in the WS documentation.

ASearch firstName, ASearch lastName

Searches in the nameFirst and nameLast columns respectively in the People table

ASearch licYear, ASearch licenceNo

City license = two digits of year (folderYear) + folder number (folderSequence)

ASearch licenceType

Searches in the SubCode column after getting the license type name and matching it against a parallel array (retrieved by a custom SQL query programmed by the AMANDA team and used in LicenceTypes.cs)

ASearch fTypes

Narrows down the search to specified professional license types ("GM", "CTR", "NEWS", "RVE") for faster searching.

ASearch VinSearch, ASearch ONPlate, ASearch CityPlate

Searches for the VIN and ON plate number, and the CityPlate in the FolderInfo table with the infocodes 40102, 40103, 40104 respectively. The InfoCodes must be specified, as the lookup column is the InfoValue column

Arrays of WSSearchCriteria objects are then constructed in different ways, depending on the search terms entered. Here they are, listed below, from lowest to highest priority, i.e. the final criterion, license number, supersedes all other criteria if data is entered here and all other fields are ignored:

1. ASearch[] search = new ASearch[] { firstName, lastName, licenceType, orgName, fTypes };  
   search = search.Where(c => c.value != null).ToArray(); //removes empty search criteria fields
2. search = new ASearch[] { CityPlate };
3. search = new ASearch[] { ONPlate }; //since license plate is unique, requested by Hongli
4. search = new ASearch[] { VinSearch }; //since VIN is unique, requested by Hongli
5. search = new ASearch[] { opNameSearch }; //operating name solo, requested by Anthony
6. search = new ASearch[] { licYear, licenceNo };

### Methods

The following Web service methods were used to retrieve records for issued professional licenses. Refer to the documentation and code for more details.

searchFolderCount  
searchFolder  
getFolderPeople  
getPeopleInfo  
getFolderFreeFormByCode – gets license type name with infocode 100  
getFolderInfoByInfoCode  
 40020 – Operating name/alias  
 40100 – make  
 40101 – model year  
   
 40102 – VIN  
 40103 – Ontario license plate  
 40104 – City license plate

getFolderAttachment  
getAttachmentContent

## Controllers

### GuardCardController.CS

Creates the objects from the models and executes the query to pass the results and pagination onto the SearchResult view.

### IndexController.CS

Simple controller for the Index page.

## Views

### GuardCard

#### Finder.cshtml

The search page

#### SearchResult.cshtml

The main page displaying the search results

### Shared

#### PersonInfoDisplay.cshtml

The individual data per record: a list of DisplayInformation is passed, and data from the Person and Card model, including the image, are manifested here per element.

#### PageLink.cshtml

Page navigation overlay

#### \_Layout.CSHTML

Navbar, header, footer

### Index

## Images

Images were previously retrieved from stored blobs on the database. They are now retrieved as base64 strings from the Web service.

License images have an attachment code of 40005.

# Business License

## Overview

Business licenses can be searched by either of two available methods: by address/location or by business name.

The application returns results of both issued and pending licenses.

## AmandaWS.CS

See Personal License section

## Property2015.aspx.cs

protected void Search\_Click(object sender, EventArgs e)

Older versions of the application encapsulated data validation and return strings. Since one instance of a Web service can be open at a time (unless async/await methods are utilized), an instance of the Amanda Service port client is opened during a valid search query and disposed after use.

A client is opened with a validated lid passed in its header (refer to previous section, or p. 29 of the WS documentation).

### Search Criteria

Refer to p. 97, Illustration: Search Web-Services portion of the Amanda Web Services documentation for the description.

**Note**: To set operator, negative, and cojuctiveOperator members, you should also set operatorSpecified, negativeSpecified, and conjuctiveOperatorSpecified Boolean values to true.

The following WSSearchCriteria objects are statically defined:

ASearch fTypes

Narrows the search to folders holding license details with business license folder types ("BROK", "CBOX", "GL", "FWKS").

ASearch expTypes

Narrows the search to folders with expiry dates no more than 14 years old (this was how it was defined in the previous version of the app).

ASearch finalDate

Searches applications with a null final date—meaning the license has been issued.

ASearch exStatus

Searches applications that do not have a status code of 7 (closed)—this was how it was defined in the previous version of the app.

ASearch notPending

Eliminates records with a status code of 14 (pending) so that they do not show up in the issued records count or results.

ASearch pendingSearch

Used in the pending records queries to search only through records with a status code of 14 (pending).

The following WSSearchCriteria objects are created to store search criteria values typed by the user:

ASearch streetSearch, civicNo, unitNo

Passes the values for the location search

ASearch bizSearch

Passes the values for the business name search, looking through the People table in the organizationname column.

Arrays of the WSSearchCriteria are then constructed in the 4 ways to search for issued and pending records by location or business name:

ASearch[] query = new ASearch[] { fTypes, expDate, streetSearch, civicNo, unitNo, finalDate, notPending, exStatus }; // address + issued

ASearch[] pendingQuery = new ASearch[] { fTypes, streetSearch, civicNo, unitNo, pendingSearch }; // pending + address

ASearch[] bizQuery = new ASearch[] { fTypes, bizSearch, expDate, finalDate, notPending, exStatus }; // business name + issued

ASearch[] pendingBiz = new ASearch[] { fTypes, bizSearch, expDate, pendingSearch }; // business name + pending

## Web Service Methods

The following Web service methods were used to retrieve records for both issued and pending licenses. Refer to the documentation and code for more details.

* searchFolderCount
* searchFolder
* getProperty
* getFolderPeople
* getFolder
* getFolderFee – this one is hidden in the documentation, but it takes arguments of an array of folderRSN int and a Boolean value, and returns an object of WSFee. This is used to pull up the outstanding fee for a pending license record. You can test it in SOAP UI.
* getFolderFreeFormByCode – gets license type with infocode 100

# Deprecated – for Reference

## Business License Methods

The methods for the old version of the application can be found in the Version History of TFS or the OldPropertyCodes.txt file in the LicenseMobile application folder.

## Connection Strings

Dev:

AMANDADV =

  (DESCRIPTION =

    (ADDRESS\_LIST =

      (ADDRESS = (PROTOCOL = TCP)(HOST = v-host)(PORT = 1521))

    )

   (CONNECT\_DATA =

      (SERVICE\_NAME = xxxx)

    )

  )

User ID: \*\*\*\*\*\*Password: \*\*\*\*\*\*\*

Alternate:

AMANDADV =

  (DESCRIPTION =

    (ADDRESS\_LIST =

      (ADDRESS = (PROTOCOL = TCP)(HOST = v-host)(PORT = 1521))

    )

   (CONNECT\_DATA =

      (SID = xxxxx)

    )

  )

In the Web.config file:

Dev database:

<add name="OracleConnectionString"   
connectionString="  
Provider=OraOLEDB.Oracle;  
 Data Source=(  
 DESCRIPTION=  
 (ADDRESS=(PROTOCOL=tcp)  
 (HOST=v-host)(PORT=1521)  
 )  
 (CONNECT\_DATA=(SERVICE\_NAME=xxxxxx)  
 )  
 );  
User Id=USER;Password=PSWD;" />