

Virginia Tech

# VLDS Web Service Interfaces

INTERNAL ONLY

## Contents

Contents .....	2
1 Overview .....	4
2 Data Adapter Web Service Interfaces .....	6
2.1 Data Adapter as Producer .....	6
2.1.1 DA_Ping-WS .....	6
2.1.2 DA_Hash_Query-WS .....	6
2.1.3 DA_Performance_Query-WS .....	6
2.1.4 DA_Generate_UEID-WS .....	7
2.2 Data Adapter as Consumer .....	8
2.2.1 LX_Ping-WS .....	8
2.2.2 SH_Ping-WS .....	10
3 Lexicon Web Service Interfaces .....	10
3.1 Lexicon as a Producer .....	11
3.1.1 LX_Ping-WS .....	11
3.1.2 LX_Metadata_Update-WS .....	11
3.1.3 LX_ValidValues_Update-WS .....	11
3.1.4 GetSchema-WS .....	11
3.1.5 GetValidValues-WS .....	12
3.2 Lexicon as a Consumer .....	12
3.2.1 CRM_Poke-WS .....	12
3.2.2 DA_Ping-WS .....	12
4 Shaker Web Service Interfaces .....	12
4.1 Shaker as a Producer .....	12
4.1.1 SH_Ping-WS .....	13
4.1.2 SH_AcceptDataPackage-WS .....	13
4.1.3 SH_ReportDataPackageStatus-WS .....	13
4.1.4 SH_ZIP_FIPS-WS .....	14
4.2 Shaker as a Consumer .....	14
4.2.1 DA_Hash_Query-WS .....	14

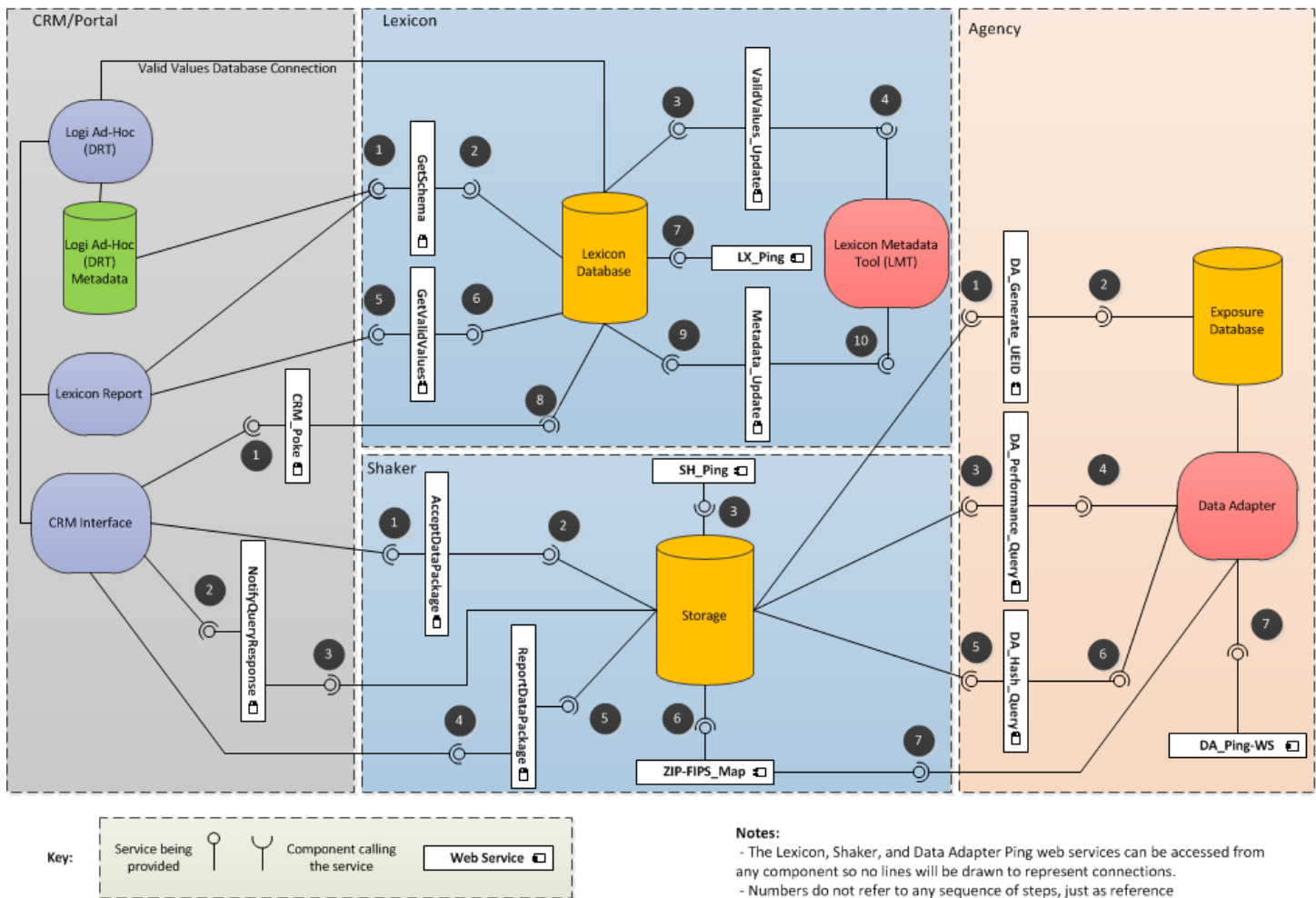
[illegible]

## 1 Overview

This document is used to explain all the web services used for communication between the Exposure Database, Data Adapter, Lexicon, Shaker, and the CRM. The web services are the backbone to the VLDS and critical in testing system connections, creating data requests, viewing metadata, executing queries, and data output. The document will show all web services produced (hosted by) and consumed by the component in the VLDS.

INTERNAL ONLY

## VLDS Web Service Interactions

**CRM/Portal Web Service Notes**

1. CRM\_Poke web service is hosted by the CRM and initiated to pull current Lexicon structure and Valid Values.
2. NotifyQueryResponse web service delivers data package responses to the CRM.
3. NotifyQueryResponse provides service which allows the Shaker to send data package responses to the CRM.

**Lexicon Web Service Notes**

1. GetSchema web service provides Metadata and top 5 Valid Values to the DRT Metadata Database and Lexicon Report.
2. GetSchema web service pulls all Metadata and top 5 Valid Values from Lexicon.
3. ValidValues\_Update web service provides Valid Values entered through the LMT.
4. ValidValues\_Update web service retrieves Valid Values from the LMT.
5. GetValidValues web service provides all Valid Values to the Lexicon Report.
6. GetValidValues retrieves all valid values from Lexicon.
7. Ping web service hosted in the Lexicon that can be called to test the connection.
8. Poke web service pulls current Lexicon structure to check for differences in Metadata/Valid Values.
9. Metadata\_Update web service provides Metadata entered through the LMT.
10. Metadata\_Update web service provides service to retrieve Metadata from the LMT.

**Shaker Web Service Notes**

1. AcceptDataPackage provides service that allows the CRM to send data packages to the Shaker.
2. AcceptDataPackage web service delivers data packages to the Shaker.
3. SH\_Ping web service exists in the Shaker and can be called to test the connection.
4. ReportDataPackage provides service that allows the CRM to request a status update for a data package.
5. ReportDataPackage web service delivers status update sent by the CRM to the Shaker.
6. ZIP-FIPS\_Map provides service to retrieve ZIP and corresponding FIPS codes from the Shaker.
7. ZIP-FIPS\_Map web service delivers ZIP and corresponding FIPS codes to the Data Adapter.

**Agency Web Service Notes**

1. Shaker calls the DA\_Generate\_UEID web service.
2. DA\_Generate\_UEID web service truncates the VLDS\_PERFORMANCE\_JOIN and VLDS\_ID\_MAPPER tables in the Exposure Database before generating random Unique Entity IDs to map to Internal IDs.
3. DA\_Performance\_Query provides service to retrieve Unique Entity IDs from the Shaker that was joined during the ID Mapper process.
4. The Data Adapter calls the Performance Query web service to gather all the Unique Entity IDs.
5. DA\_Hash\_Query provides service to retrieve the Hash Query sent through the Shaker and return the results.
6. The Data Adapter calls the DA\_Hash\_Query web service to query to the Demographic Log and return the Hashed Data to the Shaker.
7. The Ping web service exists in the Data Adapter and can be called to test the connection.

## 2 Data Adapter Web Service Interfaces

This section describes all web service interactions for the Data Adapter which is located behind the agency firewall along with the Exposure Database. All web services hosted and consumed by the Data Adapter are listed.

### 2.1 Data Adapter as Producer

The web services in this section are provided by the Data Adapter. All web services provided by the Data Adapter will be accessible over the same port.

#### 2.1.1 DA\_Ping-WS

Consumer: Shaker and/or Lexicon

Description:

This simple web service is used to test connectivity from the Shaker and/or Lexicon to the Data Adapter.

Example URL: <https://dahost:port/DataAdapterPing.svc>

The response should have HTTP status code of 200 (OK) with a Content-Type of text/plain. The content should contain the Data Adapter ID (a 3 digit integer).

#### 2.1.2 DA\_Hash\_Query-WS

Consumer: Shaker

Description:

This web service is used in the Shaker's matching process for querying hashed identifying information. The query comes in to the Data Adapter and retrieves the specific Internal IDs in the Demographic Log. Next, these Internal IDs are placed in the VLDS\_ID\_MAPPER table along with a hashed Unique Entity ID randomly populated by the Data Adapter (DA\_Generate\_UEID-WS). The hashed Unique Entity IDs are sent to the Shaker where the ID Mapping process takes place.

#### 2.1.3 DA\_Performance\_Query-WS

Consumer: Shaker

Description:

The Performance Query web service is used to gather all the Unique Entity IDs from the Shaker that was joined during the ID Mapper process. The Data Adapter then queries the exposed data sets for those IDs with help from the VLDS\_PERFORMANCE\_JOIN table. That table joins the Unique Entity IDs in the VLDS\_ID\_MAPPER table with the Internal IDs in the Demographic Log. All selected columns associated with those Internal IDs are then returned.

### 2.1.4 DA\_Generate\_UEID-WS

Consumer: Exposure Database

Description:

When this web service is called it truncates both the VLDS\_ID\_MAPPER and VLDS\_PERFORMANCE\_JOIN tables in the Exposure Database. It also generates the random Unique\_Entity\_IDs (UEID) that is populated in the VLDS\_PERFORMANCE\_JOIN table during query execution.

### ~~2.2.0 DA\_Metadata-WS~~

~~Consumer: Lexicon~~

~~Description:~~

~~The Lexicon queries the DataAdapter for extended properties metadata using this OData web service.~~

~~URLs:~~

~~To get all view metadata for schema 'DOE':~~

~~<https://dahost:port/DOE/Exposure.svc/ViewMetadata>~~

~~To get all column metadata for schema 'DOE':~~

~~<https://dahost:port/DOE/Exposure.svc/ColumnMetadata>~~

~~To get metadata for 'student' view (not including its columns) in 'SHEV' schema:~~

~~[https://dahost:port/SHEV/Exposure.svc/ViewMetadata?\\$filter=view\\_name eq 'student'](https://dahost:port/SHEV/Exposure.svc/ViewMetadata?$filter=view_name eq 'student')~~

~~To get metadata for the column 'school\_fips' of 'student' view in 'DOE' schema:~~

~~[https://dahost:port/DOE/Exposure.svc/ColumnMetadata?\\$filter=view\\_name eq 'student' and column\\_name eq 'school\\_fips'](https://dahost:port/DOE/Exposure.svc/ColumnMetadata?$filter=view_name eq 'student' and column_name eq 'school_fips')~~

~~Status: 0%~~

### 2.16.0 DA\_ValidValues-WS

Consumer: Lexicon

Description:

The Lexicon queries the DataAdapter for a specific Valid Values column or all Valid Values using this OData web service.

URLs:

To get all valid values for a schema:

[https://dahost:port/DOE/Exposure.svc/Valid\\_Values](https://dahost:port/DOE/Exposure.svc/Valid_Values)

To get all valid values for a given column:

[https://dahost:port/DOE/Exposure.svc/Valid\\_Values?\\$filter=view\\_name eq 'student' and column\\_name eq 'school\\_fips'](https://dahost:port/DOE/Exposure.svc/Valid_Values?$filter=view_name eq 'student' and column_name eq 'school_fips')

Status: 0%

## **2.26.2.2 Data Adapter as Consumer**

The web services in this section are not provided by the Data Adapter, but they are consumed by the Data Adapter from the Shaker and Lexicon.

The one common URL query string field for all of these web services is *daid* which is a 3 digit number identifying the Data Adapter making the query.

### **2.26.12.2.1 LX\_Ping-WS**

Producer: Lexicon

Description:

This simple web service is used to test connectivity from the Data Adapter to the Lexicon.

Example URL:

<https://lexiconhost:port/LexiconPing.svc?daid=102>

The response should have HTTP status code of 200 (OK) with a Content-Type of text/plain. The content should be empty.

### **2.26.2 LX\_MetadataUpdateNotify-WS**

Producer: Lexicon

Description:

This is a poke from the DataAdapter to the Lexicon to let it know that metadata has changed. It is called as part of a periodically run job on the DataAdapter whose purpose is to detect changes in the Metadata (see section 8.2). The DataAdapter stores a copy of the Metadata in the Embedded-DB. This copy should be the same as what is stored in the Lexicon. When the job runs it looks for changes between this copy and what is in the Exposure-DB. If changes are found then the job notifies the Lexicon using this web service.

The poke indicates the change scope [Schema={schemaname} | View={viewname} | Column={column name}]. The Lexicon should then query the DataAdapter's Metadata Update WS to get the latest metadata for the scope.

URL examples:

Example 1: Notify Lexicon that all metadata in the 'DOE' schema needs to be updated.  
Data Adapter requests (this web service):



<https://lexiconhost:port/Metadata.svc?daid=102&schema=DOE>

Lexicon requests (see 7.1.4):

[https://dahost:port/Exposure.svc/ViewMetadata?\\$filter=schema\\_name eq 'DOE'](https://dahost:port/Exposure.svc/ViewMetadata?$filter=schema_name eq 'DOE')

and

[https://dahost:port/Exposure.svc/ColumnMetadata?\\$filter=schema\\_name eq 'DOE'](https://dahost:port/Exposure.svc/ColumnMetadata?$filter=schema_name eq 'DOE')

Example 2: Notify Lexicon that metadata in the 'Student' view (not including its columns) needs to be updated.

Data Adapter requests (this web service):

<https://lexiconhost:port/Metadata.svc?daid=102&schema=DOE&view=Student>

Lexicon requests (see 7.1.4):

[https://dahost:port/Exposure.svc/ViewMetadata?\\$filter=schema\\_name eq 'DOE'](https://dahost:port/Exposure.svc/ViewMetadata?$filter=schema_name eq 'DOE') and  
[view\\_name eq 'student'](#)

Example 3: Notify Lexicon that metadata in the 'school\_fips' column of the 'Student' view needs to be updated.

Data Adapter requests (this web service):

[https://lexiconhost:port/Metadata.svc?daid=102&schema=DOE&view=Student&column=school\\_fips](https://lexiconhost:port/Metadata.svc?daid=102&schema=DOE&view=Student&column=school_fips)

Lexicon requests (see 7.1.4):

[https://dahost:port/Exposure.svc/ColumnMetadata?\\$filter=schema\\_name eq 'DOE'](https://dahost:port/Exposure.svc/ColumnMetadata?$filter=schema_name eq 'DOE') and  
[view\\_name eq 'student' and column\\_name eq 'school\\_fips'](#)

Status: 0%

2.26.29 — LX\_ValidValuesUpdateNotify-WS

Producer: Lexicon

Description:

This is a poke from the DataAdapter to the Lexicon to let it know that valid values have changed for an entire schema, or just a column. The Lexicon should then query the DataAdapter's Valid Values Update WS to get the latest values for the indicated scope.

The LX\_ValidValuesUpdateNotify-WS is sent during the execution of a job in the DataAdapter as described in section 8.3.

URL examples:

Example 1: Notify Lexicon to get all valid values for DOE schema.

Data Adapter requests (this web service):

<https://lexiconhost:port/ValidValuesUpdate.svc?daid=107&schema=DOE>

Lexicon requests (see 7.1.5):

[https://dahost:port/Exposure.svc/Valid\\_Values?\\$filter=schema\\_name eq 'DOE'](https://dahost:port/Exposure.svc/Valid_Values?$filter=schema_name eq 'DOE')

Example 2: Notify Lexicon to get valid values for 'school\_fips' column.

Data Adapter requests (this web service):

[https://lexiconhost:port/ValidValuesUpdate.svc?daid=107&schema=DOE&view=student&column=school\\_fips](https://lexiconhost:port/ValidValuesUpdate.svc?daid=107&schema=DOE&view=student&column=school_fips)

Lexicon requests (see 7.1.5):

[https://dahost:port/Exposure.svc/Valid\\_Values?\\$filter=schema\\_name eq 'DOE' and view\\_name eq 'student' and column\\_name eq 'school\\_fips'](https://dahost:port/Exposure.svc/Valid_Values?$filter=schema_name eq 'DOE' and view_name eq 'student' and column_name eq 'school_fips')

Status: 0%

2.26.48 — LX\_ValidValues-WS

Producer: Lexicon

Description:

A job runs periodically in the DataAdapter to check for changes in Valid Values. To do this, the job must query the Lexicon using this web service to determine the Lexicon's current Valid Values. It then compares these with the Valid Values in the Exposure DB. Differences are then communicated back to the Lexicon using the LX\_ValidValuesUpdateNotify-WS. This job is described in section 8.3.

URL example:

<https://lexiconhost:port/ValidValues.svc?daid=104&schema=DOE>

Status: 0%

2.26.55 2.2.2 SH\_Ping-WS

Producer: Shaker

Description:

This simple web service is used to test connectivity from the Data Adapter to the Shaker.

Example URL: <https://lexiconhost:port/ShakerPing.svc?daid=102>

The response should have HTTP status code of 200 (OK) with a Content-Type of text/plain. The content should be empty.

### 3 Lexicon Web Service Interfaces

The Lexicon Database hosts web services used for interacting with the Lexicon Metadata Tool (LMT), Data Request Tool, and Lexicon Report. Two web services are used for updating metadata and Valid

Values from the LMT and two are used to send structure and Valid Values to the DRT and Lexicon Report. All web services consumed by the Lexicon are listed as well.

### 3.1 Lexicon as a Producer

The web services in this section are provided by the Lexicon and have integration points with the Data Request Tool, Lexicon Report, and Lexicon Metadata Tool.

#### 3.1.1 LX\_Ping-WS

Consumer: Lexicon Metadata Tool

Description:

This simple web service is used to test the connectivity between the Lexicon Database and the Lexicon Metadata Tool. The response should have HTTP status code of 200 (OK) with a text/plain content type.

#### 3.1.2 LX\_Metadata\_Update-WS

Consumer: Lexicon Metadata Tool

Description:

This web service is used when the Lexicon Database queries the Lexicon Metadata Tool for extended properties metadata.

#### 3.1.3 LX\_ValidValues\_Update-WS

Consumer: Lexicon Metadata Tool

Description:

This web service is used when the Lexicon Database queries the Lexicon Metadata Tool for a specific Valid Values column or all Valid Values.

#### 3.1.4 GetSchema-WS

Consumer: CRM/Portal (DRT and/or Lexicon Report)

Description:

The GetSchema web service is hosted by the Lexicon and used to manipulate Lexicon changes into XML that is sent to the DRT in the CRM. The GetSchema web service sends only the database structure, Metadata, and top 5 Valid Values. There is another web service interface between Lexicon database and DRT that will notify the DRT when changes have been made to Lexicon and is referred to as the Poke Web Service (known as CRM\_Poke-WS). After the Poke web service is initiated, the DRT makes a call to the GetSchema web service to retrieve structure, Metadata and top 5 Valid Values from the Lexicon. The GetSchema web service also interacts with the Lexicon Report in the CRM to populate the database structure from the Lexicon.

### 3.1.5 GetValidValues-WS

Consumer: CRM/Portal (Lexicon Report)

Description:

The GetValidValues web service is used to extract all Valid Values from the Lexicon and populate them in the Lexicon Report within the CRM. This web service only interacts with the Lexicon Report and no other component in the CRM. The Data Request Tool retrieves all Valid Values through a database connection to the Lexicon, not through this web service.

## 3.2 Lexicon as a Consumer

The web services in this section are not provided by the Lexicon, but they are consumed by the Lexicon from the CRM and Data Adapter.

### 3.2.1 CRM\_Poke-WS

Producer: CRM/Portal (DRT & Lexicon Report)

Description:

The Poke web service hosted by the CRM is used to retrieve all Metadata and top 5 Valid Values from the Lexicon. The web service takes Lexicon updates and compares it to the current Metadata. Next it calls the GetSchema and GetValidValues web service to retrieve Metadata and Valid Values from the Lexicon.

### 3.2.2 DA\_Ping-WS

Producer: Data Adapter

Description:

This simple web service is used to test connectivity from the Lexicon to the Data Adapter.

Example URL: <https://dahost:port/DataAdapterPing.svc>

The response should have HTTP status code of 200 (OK) with a Content-Type of text/plain. The content should contain the Data Adapter ID (a 3 digit integer).

## 4 Shaker Web Service Interfaces

The Shaker hosts web services used for interacting with the Data Adapter and CRM. Two web services are used for updating metadata and Valid Values from the LMT and two are used to send metadata and Valid Values updates to the DRT and Lexicon Report.

### 4.1 Shaker as a Producer

The web services shown below are hosted by the Shaker and consist of interactions to the Data Adapter and CRM.

#### 4.1.1 SH\_Ping-WS

Consumer: Data Adapter

Description:

This simple web service is used to test connectivity from the Data Adapter to the Shaker.

Example URL: [https:// lexiconhost:port/ShakerPing.svc?daid=102](https://lexiconhost:port/ShakerPing.svc?daid=102)

The response should have HTTP status code of 200 (OK) with a Content-Type of text/plain. The content should be empty.

#### 4.1.2 SH\_AcceptDataPackage-WS

Consumer: CRM/Portal

Description:

This web service allows the CRM to send Data Packages from the researcher to the Shaker.

**The Data Package contains the following:**

- *PackageId* – Guid uniquely identifying the Data Request package
- *Name* – Human readable identifier for the Data Request
- *DataRequests* – An array of DataRequest objects representing the queries to be executed

**The Data Requests within the Data Package contain the following:**

- *DataRequestId* – Guid uniquely identifying this Data Request
- *IncludeUnmatched* – Specifies that for this data request, both matched and unmatched records should be returned
- *Name* – Human readable identifier for the data request
- *ReportID* – The Logi Report Id for the request
- *StartingAgency* – The agency whose master table is the start of the query
- *SelectedColumns* – An array of query field objects representing the columns that were selected as output for the data request
- *Filter* – A DataRequestFilterGroup object containing the filters
- *Query* – a string with the query from Logi to be used for troubleshooting purposes. This should never be shown to an agency sponsor or researcher since it will contain the internal column names.

#### 4.1.3 SH\_ReportDataPackageStatus-WS

Consumer: CRM/Portal

Description:

This web service allows the CRM to request a status update on the data package sent through SH\_AcceptDataPackage-WS. The update is sent back as a DataPackageResponse and DataRequestResponse.

**The Data Package Response contains the following:**

- *PackageId* – Guid uniquely identifying the Data Request package
- *Completed* – Boolean indicating that the package processing is completed.
- *Responses* – An array of DataRequestResponse objects representing the responses for each corresponding DataRequest

**The Data Request Response contains the following:**

- *DataRequestId* – Guid uniquely identifying the Data Request being responded to.
- *DateStarted* – The date processing was started on the data request.
- *Status* – String with status or error message.
- *FileName* – Array of strings with the file name(s) containing the results.
- *HasError* – Indicates that an error occurred when processing the data request. If true, the status should contain the error message

#### 4.1.4 SH\_ZIP\_FIPS-WS

Consumer: Data Adapter

Description:

This web service was built to translate state FIPS codes to ZIP codes. The data sets that contain ZIP code columns will be returned to the researcher instead of FIPS codes. This will save time for the researcher who doesn't have to manually translate the codes.

## 4.2 Shaker as a Consumer

The web services in this section are not provided by the Shaker. Rather they are consumed (used by) the Shaker in some way from the CRM and Data Adapter.

#### 4.2.1 DA\_Hash\_Query-WS

Producer: Data Adapter

Description:

This web service is used in the Shaker's matching process for querying hashed identifying information. The query comes in to the Data Adapter and retrieves the specific Internal IDs in the Demographic Log. Next, these Internal IDs are placed in the VLDS\_ID\_MAPPER table along with a hashed Unique Entity ID randomly populated by the Data Adapter (DA\_Generate\_UEID-WS). The hashed Unique Entity IDs are sent to the Shaker where the ID Mapping process takes place.

#### 4.2.2 DA\_Performance\_Query-WS

Producer: Data Adapter

Description:

The Performance Query web service is used to gather all the Unique Entity IDs from the Shaker that was joined during the ID Mapper process. The Data Adapter then queries the exposed data sets for those IDs with help from the VLDS\_PERFORMANCE\_JOIN table. That table joins the Unique Entity IDs in the VLDS\_ID\_MAPPER table with the Internal IDs in the Demographic Log. All selected columns associated with those Internal IDs are then returned.

#### 4.2.3 NotifyQueryResponse-WS

Producer: CRM/Portal

Description:

This web service allows the Shaker to send Data Package responses to the CRM after the initial Data Package query has gone through the Shaker. The web service compliments the SH\_ReportDataPackageStatus-WS by retrieving a response to the status update. The response is sent back as a DataPackageResponse and DataRequestResponse.

**The Data Package Response contains the following:**

- *PackageId* – Guid uniquely identifying the Data Request package
- *Completed* – Boolean indicating that the package processing is completed.
- *Responses* – An array of DataRequestResponse objects representing the responses for each corresponding DataRequest

**The Data Request Response contains the following:**

- *DataRequestId* – Guid uniquely identifying the Data Request being responded to.
- *DateStarted* – The date processing was started on the data request.
- *Status* – String with status or error message.
- *FileName* – Array of strings with the file name(s) containing the results.
- *HasError* – Indicates that an error occurred when processing the data request. If true, the status should contain the error message