Repositories

Health Data Repository 3.15 VIP Build 1

System Design Document Volume 1

(Volume 1 of 2)

Version 0.6



December 2016

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 09/06/2016 | 0.1 | Initial document creation using ProPath template v2.10 dated June 2015. | Narasa Susarla, Pat Egbert |
| 09/20/2016 | 0.2 | Updated data flow diagrams; updated Sections 1 and 2.1, updated Table 7 to include entry for logging sensitive patients, updated TRM table, added section for Service Design, consolidated Table 36. | Narasa Susarla, Pat Egbert |
| 09/28/2016 | 0.3 | Updates to Section 5 Data Design for HTH; Added Rational link to sections 1.1, 2.2, 2.3, and Appendix A.3. Updates to Sections 3.1.3.1, 3.2.1.4, and Table 15 regarding multiple VistA sites for sensitive patient logging. | Craig Wood, Pat Egbert, Narasa Susarla |
| 10/05/2016 | 0.4 | Sent for Peer Review | Pat Egbert |
| 10/12/2016 | 0.5 | Updated TRM Table 21; Table 15 – added additional verbiage; Table 28 (added RequestAndExamsRead2); Table 32 – changed from REST to SOAP for Pathways and changed template to: SensitivePatientAccessCreate1  Sent to AERB for review | Pat Egbert |
| 10/21/2016 | 0.6 | Removed Rational links in Sections 1.1, 2.2, 2.3, and Appendix A.3. Replaced with correct verbiage. | Pat Egbert |

Table of Contents

1. Introduction 1

1.1. Scope 2

1.2. User Profiles 2

2. Background 2

2.1. Overview of the System 2

2.1.1. CDS 5

2.1.2. Pathways 5

2.1.3. FPDS 6

2.1.4. Aggregate Read Service (ARS) 6

2.1.5. CDS Message Mediator 6

2.1.6. Socket Adapter 7

2.2. Overview of the Business Process 7

2.2.1. Assumptions and Constraints 7

2.3. Overview of Significant Requirements 9

2.3.1. Overview of Significant Functional Requirements 9

2.3.2. Overview of Functional Workload / Performance Requirements 9

2.3.3. Overview of Operational Requirements 10

2.3.4. Overview of Pivotal Technical Requirements 10

2.3.5. Overview of Security or Privacy Requirements 11

2.3.6. System Criticality and High Availability Requirements 12

2.3.7. Single Sign-on Requirement 12

2.3.8. Use of Enterprise Portals 12

2.3.9. Special Device Requirements 12

3. Conceptual Design 12

3.1. Conceptual Application Design 12

3.1.1. Application Context 12

3.1.2. High-Level Application Design 19

3.1.3. Application Locations 20

3.2. Conceptual Data Design 22

3.2.1. Project Conceptual Data Model 22

3.2.2. Database Information 39

3.2.3. User Interface Data Mapping 39

3.3. Conceptual Infrastructure Design 39

3.3.1. System Criticality and High Availability 39

3.3.2. Special Technology 40

3.3.3. Technology Locations 40

3.3.4. Conceptual Infrastructure Diagram 41

4. System Architecture 42

4.1. Hardware Architecture 42

4.2. Software Architecture 43

4.3. Network Architecture 44

4.3.1. CDS and Associated Services 45

4.4. Service Oriented Architecture (SOA) / ESS 45

4.5. Enterprise Architecture 46

5. Data Design 64

5.1. DBMS Files 64

5.2. Non-DBMS Files 65

5.3. Data View 65

6. Detailed Design 65

6.1. Hardware Detailed Design 65

6.2. Software Detailed Design 65

6.3. Network Detailed Design 66

6.4. Security and Privacy 66

6.4.1. Security 66

6.4.2. Privacy 67

6.5. Service Oriented Architecture / ESS Detailed Design 67

6.5.1. Service Descriptions 68

7. External System Interface Design 77

7.1. Interface Architecture 78

7.1.1. CDS Modules Overview 78

7.1.2. Mechanisms to interface CDS with external Clients 83

7.1.3. Client Interface Detailed Design 88

7.2. Interface Detailed Design 108

7.2.1. HTH Client 108

7.2.2. FtP Client 109

8. Human-Machine Interface 109

8.1. Interface Design Rules 109

8.2. Inputs 110

8.3. Outputs 110

8.4. Navigation Hierarchy 110

Appendix A. Additional Information 111

Appendix B. Sample Design Screen Shots 112

Attachment A. Approval Signatures 147

Attachment B. Signature Verification 148

**Table of Figures**

Figure 1 HDR 3.15 Clients and Interactions 4

Figure 2 HDR 3.15 Proposed Production System 13

Figure 3 CUD Process Flow for writing to HDR Database 23

Figure 4 Asynchronous Process Flow for Writing to VistA 25

Figure 5 Synchronous Write Request to VistA Process Flow 27

Figure 6 Read Request Process Flow 29

Figure 7 Get Schema Process Flow 31

Figure 8 Get Patient Identifiers Process 32

Figure 9 Query Data Process 32

Figure 10 Handle Error Process 33

Figure 11 Data Extract Process 34

Figure 12 Data Import Process 35

Figure 13 Census Read Data Model for Reporting 37

Figure 14 Example of XML report data for the model above 37

Figure 15 Production String Environment 42

Figure 16 General Support System Diagram 43

Figure 17 Repositories SOA Architecture 46

Figure 18 CDS interactions with services/components/data repositories 77

Figure 19 SOAP and REST server-side implementation, configuration of delegating objects 81

Figure 20 SOAP calls without WSRR: 82

Figure 21 Read Response Template 90

Figure 22 Patient Identification in a Filter 90

Figure 23 Entry Point Section of a Filter 91

Figure 24 eHMP XML Response with Error Section 92

Figure 25 FPDS VPR Subscribe-Get REST Web Service Interface 93

Figure 26 FPDS VPR Patient-Get REST Web Service Interface 97

Figure 27 ARS Report Filter 102

Figure 28 HTH Level of Care Report Filter 102

Figure 29 Category of Care (COC) - Total Patient Count of Non-institutionalized Care (NIC) Report Filter 103

Figure 30 Read Response Template 104

Figure 31 Example ARS XML Response 104

Figure 32 ARS XML Response with Error Section 106

Figure 33 Write Request Template 108

Figure 34 Allergies Write Request Template 112

Figure 35 ADL Data Write Template 112

Figure 36 HTH DMP Response Write Template 112

Figure 37 Write Clinical Data Response Template 113

Figure 38 Read Response Template 113

Figure 39 OP Read Response Template 114

Figure 40 Lab CH Read Response Template 115

Figure 41 HTH Surveys Read Response Template 116

Figure 42 Non-VA Medications Read Response Template 117

Figure 43 Immunization Read Response Template 118

Figure 44 Skin Test Read Response Template 119

Figure 45 Problem List Read Response Template 120

Figure 46 PSS Schematic 121

Figure 47 ADL Schematic 122

Figure 48 Appointment Read Response Template 123

Figure 49 DMP Generic Response 124

Figure 50 DMP Generic Response with Scores 124

Figure 51 TIU Domain Read Filter 125

Figure 52 Non-VA Medications Filter schematic 125

Figure 53 Immunizations Filter Schematic 126

Figure 54 Skin Test Filter Schematic 126

Figure 55 Problem List Filter Schematic 127

Figure 56 AppointmentSinglePatient Filter Schematic 127

Figure 57 RequestsandExamsSinglePatient Filter Schematic 128

Figure 58 HTH Survey Reads Request Filter 129

Figure 59 MHV Allergies Read Filter 129

Figure 60 MHV Lab Read Filter 130

Figure 61 RDI Allergies and OP Reads Data Filter 131

Figure 62 Mobile Health Appointments Read Data Filter 132

Figure 63 Error Section in Read request Response Message 133

Figure 64 JSON response with error section{"sites": [ 140

Figure 65 XML format read response 142

Figure 66 JSON response with error section{"sites": 143

Figure 67 XML response with error section 143

**Table of Tables**

Table 1 Data Stored in HDR DB 3

Table 2 HDR 3.15 High Level Functional Requirements 9

Table 3 Workload and Performance Requirements 9

Table 4 Operational Requirements 10

Table 5 Technical Requirements 11

Table 6 AITC Security Requirements 11

Table 7 Objects 14

Table 8 Interfaces Internal to OIT 18

Table 9 Externally Shared Data Stores 19

Table 10 Objects / Components to be Built or Modified 20

Table 11 Internal Data Stores 20

Table 12 Application Locations 21

Table 13 Create, Update, and Delete Process for HDR Steps 24

Table 14 Asynchronous Create and Update Process to VistA Steps 26

Table 15 Synchronous Write Request to VistA Process Flow 28

Table 16 Read Request Process Steps 30

Table 17 Data Extract Process Steps 34

Table 18 Data Import Process Steps 36

Table 19 Database Inventory 39

Table 20 Technology Location Details 40

Table 21 HDR TRM Technologies / Tools 48

Table 22 Security Control Status 66

Table 23 FPDS JMS Write Clients 68

Table 24 FPDS REST Web Service Read Clients 68

Table 25 ARS SOAP Web Service Read Clients 70

Table 26 CDS REST Web Service Clients 70

Table 27 CDS SOAP Web Service Read Clients 71

Table 28 Pathways SOAP Web Read Service Clients 72

Table 29 Socket Adapter Read Clients 74

Table 30 CDS Message Mediator Clients 75

Table 31 CDS Message Mediator Clients 76

Table 32 Pathways SOAP Web Service Write Clients 76

Table 33 CDS Data Sources 78

Table 34 Read Transactions – Domain Entry Point, Template and Filter 84

Table 35 Write Transactions – Templates 88

Table 36 Mandatory and Optional Query Parameters - Subscribe 94

Table 37 Mandatory and Optional Query Parameters – Patient Data 95

Table 38 Mandatory and Optional Query Parameters – Cancel 96

Table 39 Mandatory and Optional Query Parameters – FPDS Patient-Get 98

Table 40 ARS Report Identifiers 100

Table 41 HTH Client 108

Table 42 FtP Client 109

Table 43 Exceptions and Errors 134

# Introduction

The Repositories/Health Data Repository (HDR) system encompasses a relational database, HDR Database (HDR DB), as well as Simple Object Access Protocol (SOAP) and Representational State Transfer (REST) standards-based web service interfaces for retrieving data from HDR DB or Veterans Health Information Systems and Technology Architecture (VistA) or both HDR DB and VistA any time from within the Department of Veterans Affairs (VA) intranet. The system also includes asynchronous Java Messaging Service (JMS) interfaces for storage of data to the HDR Database any time from within the VA intranet. The web services enable VistA federation and data aggregation when data is retrieved from VistA.

Repositories services include Clinical Data Service (CDS), Pathways, Aggregate Read Service (ARS), and the Federated Patient Data Service (FPDS). The services support accepting an optional Secure Assertion Markup Language 2.0 (SAML 2.0) token on the SOAP headers for SOAP services and Hyper Text Transfer Protocol (HTTP) headers for REST services. This functionality is ready for deployment when VistA foundation is complete. All services are based on a Java Enterprise Edition (Java EE) framework and differ mainly in the data domains and exchange formats supported. CDS, Pathways, ARS and FPDS support Extensible Markup Language (XML) for data interchange. In addition to XML, FPDS also supports JavaScript Object Notation (JSON) for data interchange. JMS interfaces accept payload in Health Level Seven (HL7) format from CDS Socket Adapter and the VistA Interface Engine (VIE).

The service framework has been enhanced to support integrations with Enterprise Messaging Infrastructure (eMI) and includes integration with the eMI WebSphere Service Registry and Repository (WSRR) services and eMI messaging and routing services. The eMI WSRR service provides web service endpoint resolution to the HDR framework and replaced internally deployed Universal Description and Discovery Integration (UDDI) services. The eMI provides message routing capabilities as part of its Enterprise Service Bus (ESB) service that routes messages destined for the HDR to HDR service interfaces. Once all the VistA Data Extraction Framework (VDEF) links from VistA sites are updated to send data to eMI, the support for accepting VistA data through Vitria Interface Engine (VIE) and Socket Adapter will be phased out. Home Telehealth (HTH) clients will continue to send data to HDR through the Socket Adapter.

The HDR 3.15 VIP Build 1 release (hereafter referred to as HDR 3.15) contains the following new functionality:

* Enhancement to the Home Telehealth (HTH) client to provide census write modifications, validation and handling of census activity reports and patient satisfaction survey titles.
* Enhancement to Fix the Phones (FtP) client to support return of additional VistA User detail including the Internal Entry Number (IEN) and logging FtP access to sensitive patients

When HDR 3.15 services are deployed nationally, clients are immediately and transparently transitioned to the new versions of the deployed services. Clients using any revisions of HDR 3.x will not be impacted in any way other than the fact that there is additional functionality available as outlined above.

The HDR system along with its services (CDS, Pathways, ARS and FPDS) has additional capabilities, such as providing clinical and non-clinical data from a federation of VistA systems, as well as enabling storage and retrieval of Clinical Health Data Repository (CHDR) Allergy and Outpatient Pharmacy (OP), HTH Vitals, Activities of Daily Living (ADL VR-12) surveys, Patient Satisfaction Surveys, Disease Management Protocols (DMPs) and Census Reports.

## Scope

The Scope of the HDR 3.15 VIP Build 1 release includes the following new functionality:

* Enhancement to the Home Telehealth (HTH) client to provide census write modifications, validation and handling of census activity reports and patient satisfaction survey titles.
* Enhancement to Fix the Phones (FtP) client to support return of additional VistA User detail including the Internal Entry Number (IEN) and logging FtP access to sensitive patients

## User Profiles

The primary client base of the HDR 3.x system are software applications that are configured to access any one of the interfaces exposed by the HDR 3.x services, JMS modules, and socket adapter components. The developers of the data service-consuming applications must have knowledge about creating web-service client modules that will allow them to consume services exposed by the HDR system, as required, in order to meet their data access requirements.

The developers of client applications using the exposed JMS interfaces must have knowledge about JMS clients to place messages on the appropriate HDR JMS queues and setting up the required message properties, as required, in order to meet their data storage requirements. Similarly, consuming applications using the HDR socket adapter interface must have knowledge about interfacing with the socket clients. Any changes that are required by the data consuming applications on the HDR system are largely data-related which sometimes results in changes to the request and response schemas that are used for exchanging information.

In addition, the developers of the client applications must have knowledge of HL7 messaging.

# Background

## Overview of the System

The HDR system serves as the data service layer in the VA enterprise. The HDR system provides Create, Read, Update and Delete (CRUD) services support for data stored in the HDR DB and primarily Read service support for data stored in VistA. HDR provides the ability to Create progress notes in VistA through an HL7 interface. This release (HDR 3.15) will add the ability Create an entry in the sensitive patient access log in VistA.

The HDR system provides clinical, administrative and other data to its clients in HL7, XML and JSON formats. Clients access the HDR system via SOAP, REST, and JMS interfaces. Data returned to clients may include data from the HDR DB, VistA or both. The HDR system also supports storage of non-VistA data in the HDR DB as outlined below.

The HDR system implements the Data Federation Design Pattern (DFDP) to facilitate parallel access to the data sources and the aggregation of data retrieved from the various data sources. HDR integrates with the Identity Management System (IdM) to obtain corresponding local identifiers when a National Identifier is supplied on Read requests and uses this information to determine the VistA systems from which to extract data.

The core service framework, known as the CDS framework, is a Java EE-based stateless Enterprise JavaBean (EJB) application. The CDS framework encapsulates request processing, identity correspondence, data source location, persistence, data retrieval, data federation, error handling, audit logging, data aggregation and response generation. All services that are part of the HDR system make use of the CDS framework to satisfy client requirements. The web services that are based off of the CDS framework differ either in the data domains supported, payload types supported, data sources accessed, data transfer protocols supported or a combination thereof.

The core database component of the HDR system is HDR DB, an Oracle relational database. The following table delineates the data source and the data stored in the HDR DB.

Table 1 Data Stored in HDR DB

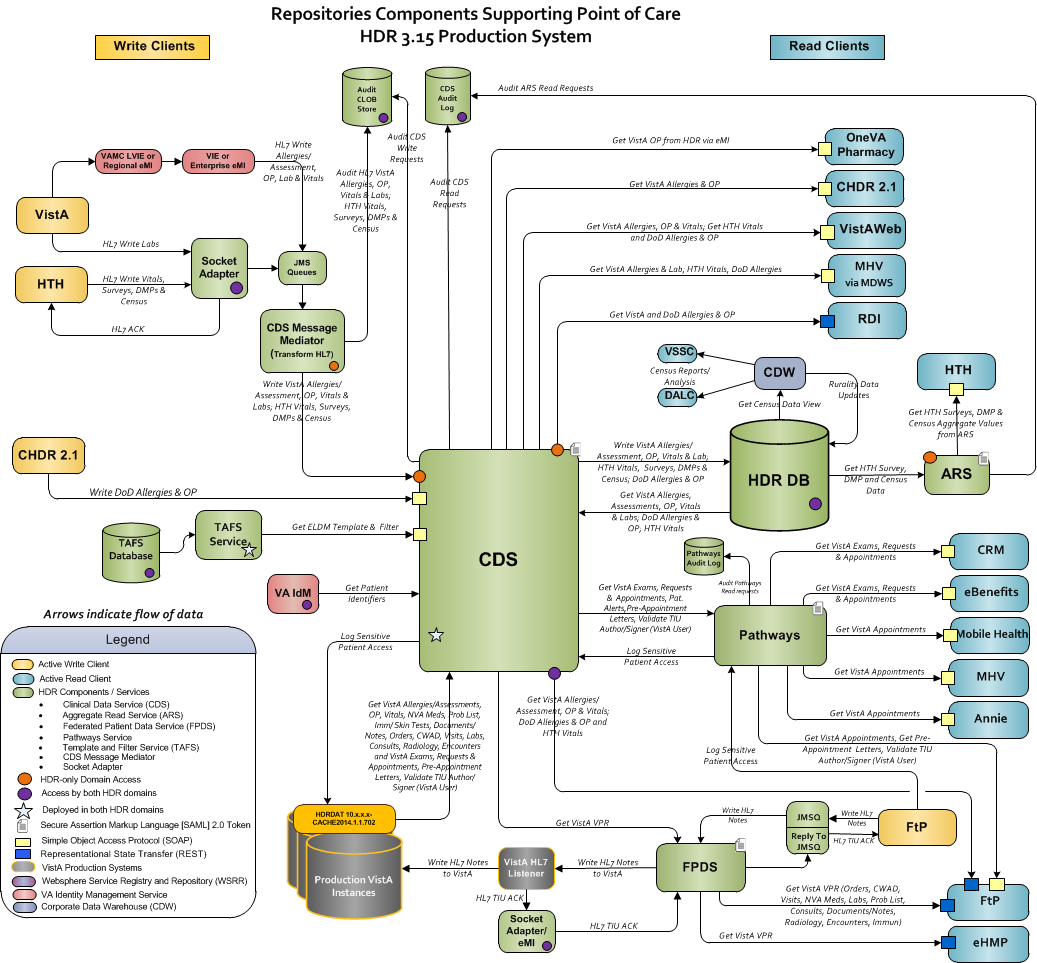
| **Data Source** | **Stored in HDR DB** |
| --- | --- |
| VistA | Allergies  Lab (Hematology and Chemistry)  Outpatient Pharmacy  Vital Signs |
| DoD via CHDR for Active Dual Consumer (ADC) patients | Allergies  Outpatient Pharmacy |
| HTH | ADL VR-12 survey responses  Disease Management Protocols (DMP)  Patient Census  Vital Signs |
| VHA Support Service Center (VSSC) data via Corporate Data Warehouse (CDW) for reporting | Patient Rurality Data |

Another core component of the HDR system is HDRDAT, which includes a set of classes for exposing VistA files through a Structured Query Language (SQL) interface. HDRDAT is a data access component deployed to each VistA instance that enables VistA to be queried via SQL. HDRDAT also includes stored procedures and custom functions that are used to obtain VistA data via Cache Object Script (COS) classes, methods, and routines. The core CDS framework interacts with HDRDAT for retrieving data from VistA.

The HDR system accepts an optional SAML 2.0 token on SOAP and REST requests. If the token is present in the request, HDR system extracts the token and passes the token to the persistent layer of the HDR framework. The CDS framework data interchange and data processing is based on XML data payloads. The XML data payload schemas, known as ‘templates’, follow the Enterprise Logical Data Model (ELDM), where available, and can be used on both requests and responses to specify data domain specific detail. The CDS framework Read interface accepts XML data, known as ‘filters’, that are transformed into query parameters for filtering data from the various data sources. The templates and filters are stored in a Template and Filter Schema in the HDR DB and can be accessed through a Template and Filter Service (TAFS). The CDS framework uses TAFS to access the templates and filters during validation and response generation processes.

The HDR system is comprised of many services and interfaces, each of which was created to meet the requirements of specific clients. The clients, services, and data stores are illustrated in Figure 1.

Figure 1 HDR 3.15 Clients and Interactions



HDR services support a request/response paradigm. All HDR SOAP services described below optionally accept a SAML 2.0 assertion on the SOAP header of a request. All REST services detailed below optionally accept a SAML 2.0 token on the HTTP header of a request. The HDR Java framework extracts the SAML 2.0 assertion from the header if present and passes it on as a parameter to the persistent layer of the HDR framework. The HDR system does not verify or parse the SAML 2.0 assertion and does not make use of the details in the SAML 2.0 assertion for any authentication or authorization at this time – this feature will be added when the VistA secure token verification Remote Procedure Calls (RPCs) are released for HDR consumption.

At the time of this writing, the VistA kernel routines that make use of the SAML 2.0 assertion are still in development and a clear access pattern to these routines is not yet established. Thus, for this release, the HDR services merely extend the service interfaces to accept an optional SAML 2.0 assertion. These extensions to HDR system are being developed to ensure minimal impact to the HDR framework and the HDR clients once the enterprise security framework is implemented and mandated across the enterprise. The SAML 2.0 assertion will not be used for authentication and authorization in HDR 3.15. All access flow processes will remain unchanged.

HDR services have been updated to perform web service endpoint resolution through eMI WSRR services. Endpoint service resolution or service lookups are typical of SOA architecture and required to resolve the location of dependent services. VA provides processes and services for registering and looking up web services through its eMI program. HDR is a SOA and depends on these services at application startup time in order to find dependent services and function properly thereafter.

### CDS

CDS is a SOAP and REST-based web service interface that supports Create, Retrieve, Update, and Delete (CRUD) operations against HDR data stores over secure Hypertext Transfer Protocol (HTTPS). CDS provides clinical data from the HDR DB, VistA or both, in XML format. CDS interfaces are generated directly from the CDS framework components using Apache CXF. CDS is accessed by OneVA Pharmacy through eMI, Home Telehealth, eHMP, CHDR for exchange of information for DoD, RDI, VistAWeb and MHV for reading and writing clinical data. CDS is deployed in both the HDR-Only WebLogic domain and the HDR + VistA WebLogic domain. Data Writes from VistA through the VistA Interface Engine (VIE) or eMI, HTH data writes, as well as RDI Read calls are handled in the HDR-only domain thus isolating these operations from any VistA-specific access issues. In addition, the FtP client uses CDS to access Outpatient Meds, VistA and DoD Allergies, and VistA and HTH Vital Signs.

### Pathways

The Pathways is another SOAP/REST web service interface accessed via HTTPS that provides administrative data from VistA in XML format. The Pathways service interface is derived directly from the CDS framework components using Apache CXF. Customer Relationship Management (CRM), eBenefits, Mobile Health, MHV and FtP use Pathways to obtain patient Appointments and Exam Request and Exam Status-related information from VistA. Pathways continues to be supported in the existing HDR + VistA WebLogic domain.

FtP uses Pathways to also obtain VistA Patient Alerts information and list of active users from a specific VistA site along with their IEN that match the specified first and last name. Additionally, FtP uses Pathways to create an entry in the sensitive patient access log at a specific VistA site.

### FPDS

#### FPDS Reads

FPDS is a REST web service accessed via HTTPS that supports reading Virtual Patient Record (VPR) data from VistA, eHMP subscription, cancellation and patient data fetches from VistA. Enhanced querying allows for specification of “amount of data” to be retrieved in any one VPR request, a form of paging larger data sets which prevents errors due to payload size. A cap of number of records at a time can be set. Subsequent queries will send a last updated token parameter to obtain only data that has changed since the last time a fetch of data was retrieved from the cache. Additionally, FPDS exposes functionality which is included in the VPR that supports a subscription model for retrieving patient data wherein additional data can be retrieved without re-execution of the VistA procedure.

The FPDS interface supports JSON and XML payloads and transforms incoming JSON requests to XML templates and filters before delegating the requests to the CDS framework for further processing. FtP uses FPDS to obtain Visits, Progress Notes/Postings, Orders, NVA Meds, Labs, Problems, Consults, Notes, Encounters, Immunizations, and Radiology from VistA using VPR.

FPDS is deployed in the existing HDR + VistA WebLogic domain. All Read calls to VPR are handled by the existing instance in the HDR + VistA WebLogic domain.

#### FPDS Writes

FPDS facilitates the routing of HL7 messages from clients to specific VistA HL7 listeners, as well as the routing of responses from each VistA HL7 listener back to the client. FPDS is a Java EE application and uses the JMS facilities of the application server to accept messages from the client.

In order to facilitate the routing of the HL7 message from the client to the appropriate VistA HL7 listener, FPDS uses the value sent inside the TIU Note MSH-6-2 (Receiving Facility), to determine the connection DNS address of the VistA HL7 listener and transmits the message for processing by the VistA HL7 package.

FPDS routes the acknowledgements, responses and errors from the VistA HL7 package to the client by placing the acknowledgements, responses and errors on a client specified response queue. If the client does not specify a response queue, FPDS places the acknowledgements, responses and errors on a default JMS response.

### Aggregate Read Service (ARS)

ARS is a SOAP web service exposed over HTTPS that provides an aggregated (report) view of HTH Survey, DMP and Census data stored in the HDR DB. ARS is deployed in the HDR-Only domain as it does not access VistA for any data.

### CDS Message Mediator

CDS Message Mediator facilitates the processing of HL7 messages sent to HDR by transforming the HL7 data into Veteran Information Model (VIM) XML data that the CDS service and framework can process, delegates the request to the CDS framework, and responds to clients in cases where acknowledgement of receipt of messages is required. CDS Message Mediator is currently used by clients to perform asynchronous create operations. CDS Message Mediator is a Java EE application that utilizes the JMS facilities of the application server to facilitate asynchronous communications. Messages are sent to CDS Message Mediator by VistA by way of the VistA interface engine infrastructure or through the HDR Socket Adapter (described below). CDS Message Mediator interacts with the CDS framework end point that uses the HDR-Only WebLogic domain and thus accesses only the HDR DB.

### Socket Adapter

The HDR Socket Adapter application is a standalone Java application that accepts Transmission Control Protocol (TCP) connections and transmission of HL7 data from HL7 Minimal Lower Layer Protocol (MLLP) clients, verifies the message source and type, and then forwards the message to CDS Message Mediator for further processing or to FPDS for routing the message back to the client. Both HTH and VistA are sending HL7 messages to the HDR via the HDR Socket Adapter. Acknowledgements, responses and errors from the VistA HL7 package are verified and sent to the HDR Socket Adapter to forward to the requesting client.

## Overview of the Business Process

HDR, together with its services, provides data services as defined by VA’s evolving Service Oriented Architecture (SOA) and can be traced to the data service layer of VA’s enterprise architecture. HDR 3.15 follows the HDR application established data service pattern and offers the capability to:

* Continue to provide a standard service interface for easy integration
* Audit and log all requests
* Integrate with Identity Management Service for Patient correlation
* Use payloads that adhere to the enterprise standards and thus establish an enterprise wide standard schema for all supported clinical and non-clinical data domains
* Monitor health of its servers to ensure continuity of operations and meet established service level agreements for every request
* Provide interfaces that accept VA enterprise supported data formats such as HL7, XML and JSON
* Adhere to modular component-based architecture that promotes reuse
* To create and update HL7 TIU Notes in the VistA system will allow easy provider communication

### Assumptions and Constraints

#### Design Assumptions

The HDR 3.15 system assumptions include the following:

* All requirements are valid and are as stated in the Change and Control Management (CCM) and Requirements Manager (RM) sections of Rational (available with authorized access only)
* All functional and non-functional requirements that were part of the previous revisions of the HDR system apply to HDR 3.15
* The new features have to be implemented within the existing framework and reuse HDR framework components where feasible
* All clients intending to create or update HL7 TIU Notes in VistA using FPDS will send the data in HL7 format and will include message properties that can be used for routing the request to Vista Logical links can be configured on VistA HL7 Listener at each VistA site and accessed from HDR services
* Responses from VistA HL7 package can be received by HDR for rerouting to the client
* All clients intending to use the HDR services for obtaining provider-specific data from VistA will integrate with Identity and Access Management (IAM) and pass a secure token for identification and authentication

#### Design Constraints

Design constraints are subject to the enterprise VA Technical Reference Model and Standards Profile (TRM/SP), which is a guide for the use of tools and programming languages. The TRM/SP includes guidance and policy regarding operating systems, database servers, and application servers. The PD ProPath process provides guidelines for software development. Any variance from these guidelines shall be approved by appropriate waiver.

The VistA HL7 package cannot communicate to a Java JMS module and hence requests must be sent to the Hl7 Listener using a socket connection. Commit Acks/Errors are returned on the initial connection and secondary Application Acks/Error responses are configured to be sent from VistA HL7 Listener through the HDR socket adapter to the appropriate response JMS Queue.

#### Design Trade-offs

* VistA VPR 1.0 JSON interface does not always return the same data fields and data as the VPR 1.0 XML interface. HDR uses VistA VPR 1.0 JSON interface for both XML and JSON read requests on FPDS in order to provide clients with the same data fields and data, irrespective of the format of data requested. HDR transforms the JSON data from VPR to XML format using XSLT when clients request XML payload responses.
* Radiology data requests from the FtP client use the VistA VPR 1.0 XML interface for both XML and JSON read requests on FPDS as the data from VPR 1.0 JSON interface for Radiology does not meet client requirements. HDR transforms the XML Radiology data from VPR to JSON format when clients request JSON payload for Radiology.
* VistA VPR does not support all the filters requested by the FtP client. HDR uses JSONPath on the response from VistA VPR to filter based on client requirements without changing VPR 1.0 JSON interface implementation. This allows existing VPR clients to use the VPR interface and enable HDR/FPDS clients to filter and obtain smaller payloads to meet their needs.
* Using a VistA HL7 interface instead of directly accessing TIU Remote Procedure Calls (RPCs) for creating and updating HL7 TIU Notes will take advantage of the VistA HL7 identity management and validation process.
* This HDR 3.14 version of the system will continue making use of materialized views that are created and populated / refreshed during off-peak hours to perform calculations and generate reports in support of HTH reporting requirements in order to mitigate the impact of calculations load.
* HDR will accept Identity Access Management and Secure Token Service (IAM-STS) secure tokens from clients on VistA data access requests and transmit it to VistA for VistA use once IAM and VistA SSOi implementation and STS are deployed to production by the IAM and VistA Kernel teams.

The rest of the design of HDR 3.14 remains the same as the previous revisions of HDR.

## Overview of Significant Requirements

### Overview of Significant Functional Requirements

The high level requirements listed in the following table represent functionality to be included in the HDR 3.15 release.

Table 2 HDR 3.15 High Level Functional Requirements

| Requirements |
| --- |
| * Enhancement to the Home Telehealth (HTH) client to provide census write modifications, validation and handling of census activity reports and patient satisfaction survey titles. * Enhancement to Fix the Phones (FtP) client to support return of additional VistA User detail including the Internal Entry Number (IEN) and provide the ability to create an entry in the sensitive patient access log |

### Overview of Functional Workload / Performance Requirements

The Repositories direction is to: (1) store and read data from HDR through HDR services deployed in the HDR-Only WebLogic domain for data domains supported by the HDR services, (2) provide access to HDR and/or VistA data for clients requiring data from VistA which may not currently be available in HDR, through HDR services deployed in the HDR + VistA WebLogic domain, and (3) provide capability to create and update HL7 TIU Notes in VistA through HDR services deployed in the HDR + VistA WebLogic domain, and (4) provide service that can be used to create an entry in the sensitive patient data access log. The HDR database and hardware procurements have been approved by Enterprise Infrastructure Engineering (EIE).

Table 3 Workload and Performance Requirements

| **ID** | **Requirement** |
| --- | --- |
| PERF 01 | The system shall respond to all RDI read requests for HDR data with an average response time of ten seconds. |
| PERF 02 | HDR shall provide 99.9% uptime availability 24/7/365. |
| PERF 03 | The response time from the time CDS receives the CHDR Read request and returns the response to CHDR shall not exceed 60 seconds 90% of the time. |
| PERF 04 | Pathways shall fulfill 99.9% of Veterans Relationship Management (VRM) Read requests in five seconds or less. (This requirement is based on matching the Read request response times for current HDR clients). |
| PERF 05 | The system shall respond to FtP read requests in 3 seconds or less in 90% of the attempts, and never more than 5 seconds, excluding large data mining activities. |
| PERF 06 | The system shall support FtP requirement of 40 Million calls per year. It is not yet known how many of these calls translate to HDR service requests. The number of calls received during the peak day of the week and the peak time of day is also not known at this time.  It is expected that the FtP call volume will increase by 20-30% due to hidden demand as these individuals are calling directly into a clinic. Nationwide calls are expected to increase by 5% for the next 5 years. |
| PERF 07 | The system shall respond to OneVA Pharmacy application outpatient pharmacy data read request and with an average response time of 5 seconds or less. |

1. Note:  HDR write operations are performed asynchronously and thus are not as time sensitive to our clients as read operations. For this reason, HDR does not have SLAs for Write operations defined with any of the HDR write clients. However, the write path is included in our capacity testing plan for testing throughput under load and is monitored in production to ensure write messages are processed in a timely manner and that there is no backlog of messages in the HDR HL7 message processing queues during peak workloads.

### Overview of Operational Requirements

Table 4 outlines the operational requirements.

Table 4 Operational Requirements

| ID | Requirement |
| --- | --- |
| OPS01 | Availability – dependent on availability of AITC – service is available when all servers in AITC are available and operational. Refer to Table 3, PERF 02. |
| OPS02 | Resource Utilization – Memory and CPU usage will vary based on the number of simultaneous requests processed and will not exceed 80% of the capacity on each application and database servers on which the Repositories application is deployed |
| OPS03 | Growth – Annual growth figures from Introscope can be provided by AITC. |
| OPS04 | Availability – dependent on availability of VistA systems and VistA HL7 Listener – HDR services will respond with errors when connection to a requested VistA system is not available. |

### Overview of Pivotal Technical Requirements

Support for all client requests shall comply with established Repositories guidelines with regard to development, implementation and release constraints, and data constraints, as well as database design and content constraints.

Other technical requirements are as listed in Table 5 below.

Table 5 Technical Requirements

| ID | Requirement |
| --- | --- |
| TR 01 | Provide a web service interface for all data access |
| TR 02 | Web service interfaces will support XML payloads by default |
| TR 03 | Support asynchronous write requests |
| TR 04 | Support XML request and response payloads |
| TR 05 | Support HL7 request and response payloads |
| TR 06 | Support JSON request and response payloads in addition to XML payloads on FPDS |
| TR 07 | Support access to VistA data in real-time |
| TR 08 | Support access to HDR data in real-time |
| TR 09 | Provide Online Transaction Processing capability for all data entry and retrieval |
| TR 10 | Provide a data cache (i.e. HDR) for VistA data in support of drug interaction checks during prescription ordering |
| TR 11 | Support data extraction by CDW from HDR DB to provision VSSC and Denver Acquisitions and Logistics Center (DALC) |
| TR 12 | Use VistA HL7 Interface for creating and updating TIU HL7 Notes in VistA |
| TR 13 | Support configuration of HL7 logical links on VistA HL7 Listener |
| TR 14 | Support HL7 request payloads for creating TIU HL7 Notes in VistA |
| TR 15 | Support eMI in parallel with VIE and Socket Adapter connections from VistA until all sites migrate to eMI |
| TR 16 | Support access to CDS service through eMI |
| TR 17 | Support SAML 2.0 assertions optionally on SOAP and REST service requests |
| TR 18 | Integrate with eMI WSRR for the purpose of Web Service Endpoint Resolution |
| TR 19 | Support SOAP interface to request the creation of an entry in the sensitive patient access log at a specific VistA site |
| TR 20 | Support data extraction from CDW to HDR DB to generate Rurality reports for VSSC |

### Overview of Security or Privacy Requirements

HDR production platforms run entirely within the Austin Information Technology Center (AITC) environment. Clients can access HDR services through Transport Layer Security (TLS). HDR uses a whitelist to ensure only authorized applications access HDR services. AITC provides the security requirements:

Table 6 AITC Security Requirements

| ID | Requirement |
| --- | --- |
| 01 | All sensitive data (i.e. PHI/PII, passwords) being transmitted must be encrypted. |
| 02 | Only VA authorized users may access VA-owned equipment used to process VA information or access VA processing services. |
| 03 | In accordance with DCO guidance, all VA/DCO personnel accessing information systems (IS) must read and acknowledge their receipt and acceptance of the VVA National Rules of Behavior (ROB) or VA Contractor’s ROB prior to gaining access to any VA IS or sensitive information. The rules describe user responsibilities and expected behavior with regard to information and information system usage.  These rules are included as part of the annual security awareness training. |
| 04 | The organization must conduct an annual Privacy Threshold Assessment (PTA) and Privacy Impact Assessment (PIA) on the IS in accordance with the Office of Management and Budget (OMB) policy. |

### System Criticality and High Availability Requirements

HDR is classified as ‘*Mission Critical’* by AITC. AITC will support an RTO/RPO of 12 hours/2 hours for HDR. The HDR requirement for RTO/RPO is 2 hours /15 Minutes. This will require a classification of ‘*Mission Critical Special* ’. HDR Tech Refresh will address the RTO/RPO ‘*Mission Critical Special’* requirement. HDR Tech Refresh is scheduled to occur in FY17. Refer to PERF 02 uptime requirement in Table 3.

### Single Sign-on Requirement

HDR functions as a back end system and does not include any user interfaces. HDR accepts SAML token on all its service interfaces.

### Use of Enterprise Portals

Not Applicable.

### Special Device Requirements

Not Applicable.

# Conceptual Design

## Conceptual Application Design

This section provides the conceptual design of the application that is being produced by this project.

### Application Context

Figure 2 represents the context in which the HDR 3.x application exists. The green components shown in this figure represent both Writes and Reads in the HDR 3.x system. Refer to Tables 7, 8, and 9 which describe the components of the figure below. The HDR-Only Access points are delineated with an orange circle as defined in the Legend.

Figure 2 HDR 3.15 Proposed Production System

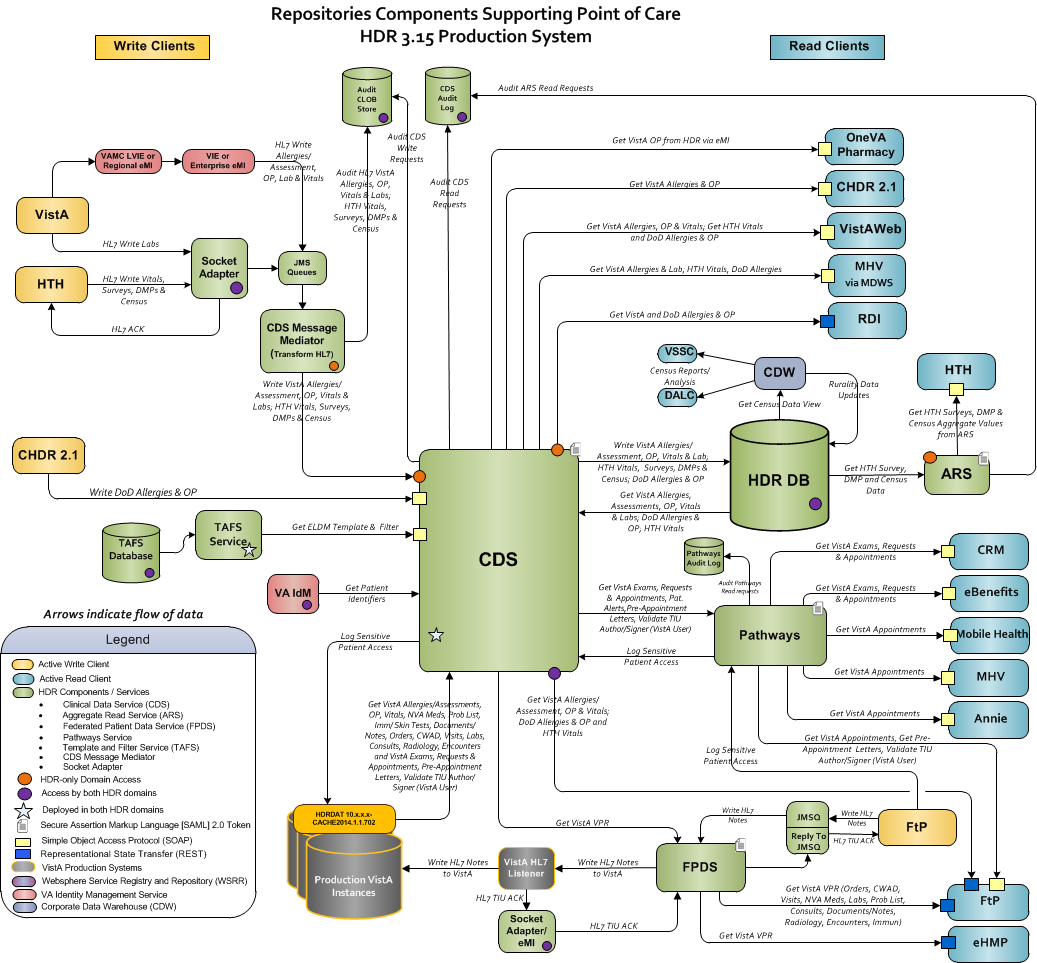


Table 7 Objects

| Name | Description | Interface Name | Interface System |
| --- | --- | --- | --- |
| **Write Clients** | | | |
| Synchronous CDS Clients – XML Messages | | | |
| CHDR | CHDR sends allergy and outpatient pharmacy data over a web service interface for active dual consumers to HDR 3.x system. | createOrUpdateClinicalData | CDS |
| Synchronous CDS Framework Clients – XML Messages | | | |
| Message Mediator | MessageMediator reads HL7 messages off of its queues, transforms them to XML format using XSL and calls CDS framework for storing the data in HDR DB. Responses from CDS are placed on a specified response queue or on a default ‘reply-to’ queue. CDS Message Mediator is deployed in the HDR-Only WebLogic domain and connects to CDS 3.x that is deployed in the HDR-Only WebLogic domain. | createOrUpdateClinicalData | CDS |
| Synchronous Pathways Clients – XML | | | |
| FtP | FtP requests Pathways service to create an entry in the sensitive patient access log. | createData | Pathways |
| Asynchronous FPDS Clients – HL7 Messages | | | |
| FtP | FtP sends ‘TIU Notes’ information for storing in VistA in HL7 format asynchronously through JMS modules defined on the FPDS service interface. Success indicator or an error message in case of failures during the storage operation on VistA will be sent back to FtP over a preconfigured queue. The JMS queues are configured in the HDR+VistA domain and uses the HDR +VistA FPDS endpoint. | send (see JMS API for creating client connections to JMS Qs) | JMS – FPDS FtP Q |
| MessageMediator Clients – HL7 Messages | | | |
| SocketAdapter | SocketAdapter places the messages it receives on an appropriate MessageMediator Queue for processing and storage | send (see JMS API for creating client connections to JMS Qs) | JMS - MessageMediator Q |
| VistA/VIE | Allergy and Outpatient Pharmacy data from VistA is transmitted to VIE that then places the HL7 messages on HDR 3.x JMS queues for processing and storage in HDR DB. The JMS Queues are configured in the HDR-Only WebLogic domain and use the HDR-Only CDS endpoint. | send (see JMS API for creating client connections to JMS Qs) | JMS - MessageMediator Q |
| VistA/eMI | Allergy, Outpatient Pharmacy, Vitals, and Laboratory data from VistA is transmitted to eMI that then places the HL7 messages on HDR 3.x JMS queues for processing and storage in HDR DB. The JMS Queues are configured in the HDR-Only WebLogic domain and use the HDR-Only CDS endpoint.  TIU HL7 Create Note acknowledgements and errors from VistA TIU HL7 listener are transmitted to eMI that then places the acknowledgements and errors on HDR 3.x JMS DefaultHL7Ack queue for processing and forwarding them to the appropriate client. | send (see JMS API for creating client connections to JMS Qs) | JMS - MessageMediator Q |
| Socket Adapter Clients – HL7 Messages (TCP/IP Socket connections) | | | |
| HTH | HTH uses HDR 3.x TCP/IP socket interface to send Vital signs data from HTH devices, Survey responses, DMPs and Census reports in HL7 message to HDR 3.x system via the Socket Adapter and CDS Message Mediator to the HDR-Only CDS endpoint. Acknowledgements from HDR 3.x will be sent back to HTH over preconfigured ‘reply-to’ queue. | (Non)BlockingConnection write  (Non)BlockingConnection receive  (See XSocket API documentation for details) | XSocket – SocketAdapter port |
| VistA Lab | VistA Lab package uses HDR 3.x TCP/IP socket interface to transmit Lab Chemistry and Hematology Result messages in HL7 format to HDR 3.x system in the HDR-Only WebLogic domain. | NonBlockingConnection write  (See XSocket API documentation for details) | XSocket – SocketAdapter port |
| VistA TIU HL7 Listener | VistA TIU HL7 Listener uses HDR 3.x TCP/IP socket interface to send Create Note AA and AR messages to HDR system. These messages are placed on the appropriate client queue in the HDR system based on the detail in the MSH 3 segment. | (Non)BlockingConnection write  (Non)BlockingConnection receive  (See XSocket API documentation for details) | XSocket – SocketAdapter port |
| **Read Clients – Synchronous Clients** | | | |
| ARS Clients – XML | | | |
| HTH | HTH accesses ARS web service interface available in the HDR-Only WebLogic domain using HTTP/SOAP for requesting and receiving aggregate reports on Survey Responses, DMPs and Census reports from HDR 3.x system. | readAggregateData | ARS |
| CDS Clients – XML | | | |
| CHDR | CHDR uses CDS 3.x web service interface on HTTP(s)/SOAP to request for and receive VistA allergies and Outpatient Pharmacy data for active dual consumers. | readClinicalData1 | CDS |
| MHV | MHV uses CDS 3.x web service interface on HTTP(s)/SOAP to request for and receive VistA allergies, HTH Vitals and Lab Chemistry and Hematology results data. | readClinicalData1 | CDS |
| RDI | RDI connects to CDS endpoint in the HDR-Only WebLogic domain using HTTP/REST to obtain CHDR and VistA allergies and outpatient pharmacy data stored in HDR DB for drug checks during prescription ordering process. | readClinicalData1 | CDS |
| VW | VS uses CDS 3.x web service interface on HTTP(s)/SOAP to request for and receive CHDR allergies and HTH Vitals data. | readClinicalData1 | CDS |
| FtP | FtP uses CDS to retrieve outpatient medications, allergies (both VistA and DoD) and vital sign observations (both VistA and HTH) from HDR. | readClinicalData1 | CDS |
| OneVA Rx/eMI | OneVA Pharmacy uses CDS through eMI to retrieve Outpatient medications from HDR. | readClinicalData1 | CDS |
| CDS Framework Clients – XML – POJO | | | |
| FtP | FtP uses the CDS framework to extract Progress Notes/Postings, Orders and Visit information using VPR 1.2 JSON API and Radiology information using VPR 1.2 XML API from VistA. | readClinicalData1 | CDS |
| Pathways | Pathways uses the CDS framework to extract non-clinical data from VistA. | readData | CDS |
| FPDS Clients – HTTP(s)/REST - JSON | | | |
| eHMP | eHMP uses FPDS REST interface to request VPR data from VistA. | Get | FPDS |
| FtP | FtP uses FPDS REST web service interface to request Progress Notes/Postings, Orders, Consults, Encounters, Immunizations, Labs, NVA Meds, Problems, and Visit information from VistA VPR. | Get | FPDS |
| FPDS Clients – HTTP(s)/REST - XML | | | |
| FtP | FtP will use FPDS REST web service interface to request Radiology information from VistA VPR. | Get | FPDS |
| Pathways Clients – HTTP(s)/SOAP - XML | | | |
| CRM | CRM uses Pathways SOAP web service interface to obtain patient Appointments as well as C&P Exam Request and Exam Status information in order to deter the status of their claims | readData | Pathways |
| eBenefits | eBenefits uses Pathways SOAP web service interface to provide upcoming Appointments as well as C&P Exam Request and Exam Status information to its users | readData | Pathways |
| FtP | FtP uses Pathways SOAP web service interface to provide Appointments, Pre-Appointment Letters, List of active users and Patient Alerts information from VistA. | readData | Pathways |
| MHV | MHV will use Pathways SOAP web service interface to notify clients of their upcoming Appointments | readData | Pathways |
| MobileHealth | MobileHealth uses Pathways SOAP web service interface to notify clients of their upcoming Appointments | readData | Pathways |
| TAFS Clients – HTTP(s)/SOAP - XML | | | |
| CDS | TAFS is an internal service used by CDS service to access schema files that represent request filters and response templates in its read and write processing for validating the payloads and requests and for generating the responses. TAFS is deployed in both the HDR-Only WebLogic domain, as well as in the existing HDR + VistA WebLogic domain. | getSchema | TAFS |
| HDRDAT Clients – JDBC | | | |
| CDS | CDS uses HDRDAT, an SQL projection layer on VistA for extracting data from VistA. In addition to SQL tables that are SQL projections of VistA FileMan Files, HDRDAT also consists of extension tables that include data columns for improved cross-table performance and stored procedures that wrap up calls to VistA APIs. Services from the HDR-Only WebLogic domain do not access HDRDAT. | HDRDAT | HDRDAT/VistA |
| Extract Client - SQL | | | |
| CDW | CDW will login and extract HTH census and survey data from HDR using SQL. Extracted data from HDR will be used by VSSC and DALC for analytic purposes. HDR will restrict access to CDW for specific tables. | Oracle | HDR |
| Import Client - SQL | | | |
| CDW | CDW will login and insert HTH rurality data into HDR using SQL. Imported data will be used by the HDR ARS service in support of HTH requirements for reporting on Rural Average Daily Census activity. | Oracle | HDR |

Table 8 Interfaces Internal to OIT

| Name | Related Object | Input Messages | Output Messages | External Party |
| --- | --- | --- | --- | --- |
| createOrUpdateClinicalData  createClinicalData  updateClinicalData | CDS | Data Domain specific XML that consists of data specific for each domain such as allergy, vitals, census, etc. The XML follows domain specific template schemas – see Appendix B for details on the templates | Contains the unique active recordId from HDR DB against which the data is inserted or updated | CHDR  HTH  eHMP  VistA |
| readAggregateData | ARS | Filter data that is used in SQL to filter and query the data sources – see Appendix A for details on the filters | Aggregate information on HTH ADL and PS surveys, DMPs and Census’s stored in HDR DB | HTH |
| readClinicalData | CDS | Filter data that is used in SQL to filter and query the data sources – see Appendix A for details on the filters | Data domain specific XML that consists of data specific to the requested domains. The XML follows domain specific template schemas – see Appendix A for details on the templates | CHDR  HTH  MHV |
| readData | Pathways | Filter data that is used in SQL to filter and query the data sources – see Appendix A for details on the filters | Data domain specific XML that consists of data specific to the requested domains. The XML follows domain specific template schemas – see Appendix A for details on the templates | CRM  eBenefits  MobileHealth |
| GET | FPDS | Path and query parameters that are used to determine the data being requested | JSON object that consist of VPR data as per the request | eHMP |

Table 9 Externally Shared Data Stores

| Name | Data Stored | Owner | Access |
| --- | --- | --- | --- |
| VistA | VistA deployed at 130+ VA sites stores all operational health and administrative data. | VAMC | Reads;  Writes for HL7 TIU Notes |
| VSSC | VA’s Corporate Data Warehouse (CDW), which is SQL Server | CDW | Read for customers; Read/Write for HDR |
| DALC | VA’s CDW, which is SQL Server | CDW | Read for customers; Read/Write for HDR |

### High-Level Application Design

Refer to Figure 2 above and associated tables 7, 8, and 9 for detailed descriptions of the HDR clients and associated components.

Table 10 Objects / Components to be Built or Modified

| ID | Name | Description | Service or Legacy Code | External Interface Name | External Interface ID | Internal Interface Name | Internal Interface ID | SDP Section 1 & 2 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Pathways | Return list of active users from specified VistA site that match the specified first and last name | Pathways/CDS | readData |  | getUsers |  |  |
| 2 | Pathways | Create an entry in the sensitive patient access at the specified VistA site using the specified user, patient and reason information | Pathways/CDS | createData |  | createSensitivePatientDataAccessLogEntry |  |  |
| 3 | ARS | Read Rurality data | ARS | readData |  | readRuralityData |  |  |

Table 11 Internal Data Stores

| ID | Name | Data Stored | Steward | Access |
| --- | --- | --- | --- | --- |
| 1 | HDR | Patient rurality data | ARS | Read |

### Application Locations

All HDR components and database, except for HDRDAT, will be hosted at AITC. HDRDAT is hosted on the Cache systems that hosts VistA. HIDU push is used to push the HDRDAT changes to VistA systems at the Veterans Administration Medical Centers (VAMCs) in production.

Table 12 Application Locations

| Application Component | Description | Location at Which Component is Run | Type |
| --- | --- | --- | --- |
| HDRDAT | SQL Projection layer that includes relational representation of VistA FileMan files, extension tables that improve performance of cross-table joins and stored procedures that wrap VistA Application Programming Interface (API) calls. | VAMCs – InterSystems Cache | Data Layer |
| HDR DB | Oracle Relational Database that houses data from ‘write’ clients identified in Figure 9. | AITC – Oracle RAC | Data Layer |
| All components except for HDRDAT | Components that are comprised of the HDR 3.x core CDS framework and all services and interfaces that interact with CDS framework | AITC – WebLogic cluster | Service Layer  Application Layer |
| VistA TIU HL7 configuration components | IRM staff at each VistA site will need to configure the following HL7 components:   1. Sending Application (HDRFTP) – This identifies FTP as the client 2. Receiving Application (TIUHL7)– The assumption is that IRM will use the existing TIU receiving application 3. HL7 Protocols – set up new MDM events and subscriber for HDR 4. Logical Link – IRM staff  will use the existing HDR logical link in the MDM event | VAMCs intersystems | Interface Layer |

#### Application Users

The HDR 3.x system does not have any direct human users. All access to its services and interfaces is through client applications. The Users interact with the client applications which in turn uses the HDR 3.x system to satisfy data requirements to meet the needs of the users. The User roles are determined by the client applications. The HDR 3.x system does not distinguish access based on user roles and determines the data to be returned based on the information specified in the request filters. Similarly, when storing data to HDR, the system uses the information specified in the templates to identify the HDR DB table in which to store the data. Refer to Tables 7-9 which illustrate the CDS application use at current clients outlined in Section 1 Introduction.

HDR supports writing data to VistA by providing Create and Update functionality for TIU HL7 Notes in VistA. The client applications specify the details of the data domain to be created or updated along with user details in an HL7 message. HDR passes the message through to VistA using the VistA HL7 Interface. The VistA HL7 package will identify and validate user information in the HL7 message prior to storing the specified data in the appropriate VistA data domain.

HDR will add support that will enable clients to create an entry in VistA DG Security log based on information specified in the payload, including the sensitive record flag (Y or N). Clients will pass in a list of all the required information per site for logging the sensitive patient access and the HDR Java service layer will call the VistA ‘DG SENSITIVE RECORD ACCESS’ RPC to log the information at each site.

## Conceptual Data Design

This section details the conceptual data model for the HDR Database. VistA is also used by the HDR system to support some of the client requirements.

1. The VistA data design and data model are not in the scope of this SDD and will not be elaborated here.

### Project Conceptual Data Model

#### Create, Update and Delete Process

The Create, Update, and Delete (CUD) Process illustrated in Figure 3, together with the Read Process illustrated in Figure 5, form the backbone of the CDS framework in the HDR system. Different aspects of this framework are exposed through different interfaces by the services. The CUD operations and Read requests that originate from RDI are handled by the HDR-Only WebLogic Domain.

All aspects of this framework (CRUD) are exposed by the CDS 3.15 service through a stateless synchronous interface. The synchronous interface is available to clients on the HDR + VistA WebLogic domain. The Create and Update aspects of the framework are also exposed through an asynchronous interface on the CDS 3.15 service via CDS Message Mediator and deployed in the HDR-Only WebLogic domain.

An asynchronous interface on FPDS and is available to clients on the HDR + VistA WebLogic domain. The Pathways and FPDS services expose the Read aspects of the framework through a stateless synchronous interface that is available on the HDR + VistA WebLogic domain. Pathways service also exposes Write aspects of the framework through a stateless synchronous interface on the HDR + VistA domain to enable creation of an entry in the sensitive patient access log in VistA sites.

#### Create, Update, and Delete Process against HDR Database

The Write operations against the HDR database break down to CUD operations, as illustrated in Figure 3 CUD Process Flow for writing to HDR Database . Write processing creates an XML write template wrapper around the JSON payload on FPDS, as well as the transformation of the XML acknowledgement or error message to JSON.

Figure 3 CUD Process Flow for writing to HDR Database



The main aspects of the Write operations flow include the following steps noted in Table 13.

Table 13 Create, Update, and Delete Process for HDR Steps

| Step | Description | Alternate Process |
| --- | --- | --- |
| 1 | Once the HDR 3.x system receives the Write request, either synchronously through the Web service or asynchronously on a JMS queue that is monitored by an MDB, an audit of the request is created in the HDR DB. | An unsuccessful audit causes an error to be logged to the HDR DB and an error response is returned to the client that initiated the request indicating the cause of error. |
| 2 | Upon successful auditing, the system validates the request against the template schema that corresponds to the template ID specified in the Write request. | The system attempts to obtain the template schema from the HDR template cache. If the schema does not exist in the cache, a call is made to TAFS to retrieve the schema into the cache. If the schema could not be located in Template and Filter Service (TAFS), an error is logged in the HDR DB and an error response is returned to the client that initiated the request indicating the cause of error.  Unsuccessful validation causes an error to be logged to the HDR DB and an error response is returned to the client that initiated the request indicating the cause of error. |
| 3 | Upon successful validation of the Write request, the database operation is executed through a Hibernate layer. | Unsuccessful execution of the Insert/Update/Delete operation on the database causes an error to be logged to the HDR DB and an error response is returned to the client that initiated the request indicating the cause of error. |
| 4 | Once the database operation is executed successfully, a success response is generated. In the case of a Create request, the success response includes the record identifier of the newly created record in the database. The success response is returned to the client that initiated the request. |  |

#### Asynchronous Create and Update Process against VistA

Asynchronous Write operations against VistA through HDR provide the ability to create and update HL7 TIU Notes in VistA through the VistA HL7 Listener. See Figure 4 Asynchronous Process Flow for Writing to VistA. When a TIU HL7 create or update message from a client arrives on the JMS Queue associated with FPDS service, the HDR system uses routing information contained in the MSH segment of the message to connect to the appropriate logical link established at a VistA site and transmits the message on the connection for further processing by the VistA HL7 Listener at the VistA site. The HDR system receives the acknowledgements, responses or errors from VistA and sends them back to the client.

The HDR system thus acts only as a broker that passes the messages to the appropriate end points and facilitates creating TIU HL7 Notes at any VistA system.

Figure 4 Asynchronous Process Flow for Writing to VistA



The main aspects of the asynchronous HL7 Write operations flow to VistA includes the following steps noted in Table 14.

Table 14 Asynchronous Create and Update Process to VistA Steps

| Step | Description | Alternate Process |
| --- | --- | --- |
| 1 | Once the HDR 3.x system receives the HL7 TIU Write request asynchronously on a JMS queue that is monitored by an MDB, a connection is established with a logical link at the VistA site specified in the HL7 message properties. | An unsuccessful connection causes an error to be logged to the HDR DB and an error response is returned to the client that initiated the request indicating the cause of error. |
| 2 | Upon successful connection, the system transmits the request to the VistA HL7 Listener on the established connection and waits for an acknowledgement or error from the VistA system. | An acknowledgement error is sent back to the client if MSH Validation fails. |
| 3 | If the initial MSH Validation passes then a secondary App ACK or App Error is received by the HDR System once VistA completes processing the request. | The Application acknowledgement or error is sent back to the client without any modifications. |

#### Synchronous Write Process to VistA

Synchronous Write process to VistA is delivered through Pathways service and provides the ability to create an entry in the sensitive patient access log at a single VistA site based on information provided in the payload. See Figure 5 Synchronous Write Request to VistA Process Flow. Note that this flow depends on VistA RPCs to create an entry in the sensitive patient access log at the specified VistA site. The Pathways Write support consists of a synchronous, stateless web service interface that uses information provided in the Write request template to identify the VistA site at which the entry should be created. The Write request template also contains details required for creating an appropriate entry in the sensitive patient access log.

Once the VistA operation is executed successfully, a response is returned to the client in the form of a status and status description. The status reflects the action taken by the VistA RPC for each site:

0 = patient is not sensitive

1 = patient is sensitive, user has security access and logs an entry in DG SENSITIVE RECORD ACCESS

2 = patient is sensitive, user does not have security access

3 = user is trying to access own patient record which causes a bulletin to be sent to DG Security group

4 = patient is sensitive but the user is missing the SSN in the user profile.

-1 = validation error

Figure 5 Synchronous Write Request to VistA Process Flow



The main aspects of the Synchronous Write to VistA process flow are included in Table 15 below.

Table 15 Synchronous Write Request to VistA Process Flow

| Step | Description | Alternate Process |
| --- | --- | --- |
| 1 | Once the HDR 3.x Pathways system receives a list of required information in the Write request for logging sensitive patient access synchronously at multiple sites through the web service, an audit of the request is created in the HDR DB. | An unsuccessful audit causes an error to be logged to the HDR DB and an error response is returned to the client that initiated the request indicating the cause of error. |
| 2 | Upon successful auditing, the system validates the request against the template schema that corresponds to the template ID specified in the Write request. | The system attempts to obtain the template schema from the HDR template cache. If the schema does not exist in the cache, a call is made to TAFS to retrieve the schema into the cache. If the schema could not be located in Template and Filter Service (TAFS), an error is logged in the HDR DB and an error response is returned to the client that initiated the request indicating the cause of error.  Unsuccessful validation causes an error to be logged to the HDR DB and an error response is returned to the client that initiated the request indicating the cause of error.  There is a potential for a partial success/failure in the event that when multiple sites are queried, some sites may work successfully, while other sites may be unavailable. In this case, the error message back to the client indicates the sites where the audit could not be performed so that this could be used to govern which sites they try to read data from for the sensitive patient. |
| 3 | After the request has been validated, the VistA sites at which the sensitive patient data access log entries need to be made is extracted from the request payload and a connection is made to the appropriate VistA systems. | If a connection to the site could not be made, an error is logged in the HDR DB and an error response is returned to the client that initiated the request indicating the cause of error. |
| 4 | Once successful connection is made to VistA, the application calls for the stored procedure to be executed in HDRDAT. This will in turn call the DG SENSITIVE RECORD ACCESS in each applicable VistA site. | Unsuccessful execution of the stored procedure or RPC causes an error to be returned to the CDS Java framework which will create an error response encapsulating the error and an error response is returned to the client that initiated the request indicating the cause of error. |
| 5 | Once the VistA operation is executed successfully, a response is returned to the client in the form of a status and status description. The status reflects the action taken by the VistA RPC for each site. See Section 3.2.1.4. |  |

#### Read Request Process

The HDR 3.15 Read request processing remains the same as is HDR 3.14 and is illustrated in Figure 6 Read Request Process Flow. The HDR Read support consists of a synchronous, stateless web service interface that uses information provided in the Read request to identify the data to be sent back to the client in the response

Figure 6 Read Request Process Flow



The main aspects of the Read operations flow include the following steps noted in Table 16.

Table 16 Read Request Process Steps

| Step | Description | Alternate Process |
| --- | --- | --- |
| 1 | Once the HDR 3.x system receives the Read request synchronously through the web service, the application determines whether transformations are required prior to processing the message. If transformations of the request are required, the appropriate transformations are applied to the incoming request. | An unsuccessful transformation causes an error to be logged to the HDR DB and an error response is returned to the client that initiated the request indicating the cause of error. |
| 2 | Upon successful auditing, the system loads the filter schema that corresponds to the filter ID specified in the Read request from a local filter cache managed by the application and validates the filter information in the request against the filter schema. | If the filter does not exist in the filter schema, the filter is loaded from the TAFS. CDS uses an eMI WSRR registry to locate the TAFS service. TAFS is invoked with the filter ID to load the filter into the application filter cache. If the filter schema could not be located in TAFS or if the TAFS web service is unavailable, an error is logged in the HDR DB and an error response is returned to the client that initiated the request indicating the cause of the error. |
| 3 | After the request filter has been validated against the filter schema, Patient information is retrieved from the filter. If the filter contains a resolvable national identifier, the VA Identity Management service is invoked to get the identifiers associated with the national Patient ID. | If there are no patient identifiers in the request or if the eMI WSRR or IdM services are unavailable, or if the Patient’s national identifier is not known by the VA IDM Service, an error is logged in the HDR DB and an error response is returned to the client that initiated the request indicating the cause of the error. |
| 4 | Once all patient identifiers have been resolved by the system, the requested data type requests along with these identities are utilized to determine the sources that should be queried to retrieve the requested patient data. | If no data sources are available for the requested data type or patient, an error is logged in the HDR DB and an error response is returned to the client that initiated the request indicating that no data could be found. |
| 5 | Once data sources are determined, connections are made to each source in parallel to retrieve the information relevant to the request. | If a connection to one or more of the sites could not be made, an error is logged in the HDR DB and an error is added to the response indicating that the response is incomplete. |
| 6 | Once all queries have completed, the response template is loaded from the internal application template cache and the results from all data sources are aggregated and sequenced per the template schema to form a single response. | If eMI WSRR or TAFS services are unavailable, an error response is returned to the client that initiated the request indicating that the response could not be created. If the response template schema corresponding to the request could not be loaded from TAFS, an error is logged in the HDR DB and an error response is returned to the client |
| 7 | After the response message has been created, information about the response is audited to the HDR DB. | If auditing of the request fails, an error is logged in the HDR DB and an error response is returned to the client |
| 8 | If further transformation of the request is required, the appropriate transformations are applied to the response message prior to returning the response message back to the client. | If the transformation fails, an error is logged in the HDR DB and an error response is returned to the client indicating a transformation error. |

#### Get Schema Process

The ‘Get’ Schema is a sub-process used in both CUD and Read operations. Figure 7 Get Schema Process Flow illustrates the process which spans both the TAFS service and its clients (CDS and Pathways). The process includes checking for the schema in the parent cache and only calling out to the TAFS service if the schema does not exist in the cache as seen in Figure 7. Errors that occur due to unsuccessful connection to the service or inability to locate the schema are handled by the parent process.

Figure 7 Get Schema Process Flow



#### Get Patient Identifiers

The ‘Get Patient Identifiers’ illustrated in Figure 8 Get Patient Identifiers Process, is a sub-process of the Read process that extracts identifiers from the request filter. In case an Integration Control Number (ICN) is specified in the filter, the VA IdM service is used for obtaining corresponding identifiers. Any exclusion specified by the client in the filter are removed from the corresponding identifiers list. This process returns “resultant identifiers” that are used to determine the sources of data and for querying data from the data sources.

Figure 8 Get Patient Identifiers Process



#### Query Data

Query data is another sub-process of the Read process and is illustrated below in Figure 9 Query Data Process. Based on the resultant identifiers and the data that needs to be obtained, appropriate connections are made to the various data sources in parallel to retrieve the required information. Depending on the data domain, either a stored procedure or an SQL query is executed and the various responses from all the databases queried are returned to the Read process for aggregation, sequencing and transformation.

Figure 9 Query Data Process



#### Handle Errors

Figure 10 Handle Error Process illustrates the details of the Handle Errors process, which is an over-arching process that is used by all processes, except audit processing, to log any exceptions that occur during processing and generate an appropriate error response element or document to be included in the response.

Figure 10 Handle Error Process



#### Data Extract Process

The data extract process illustrated in Figure 11 Data Extract Process is available directly on the HDR database. A “view” to the HDR tables limits access from the Corporate Data Warehouse (CDW) for extracting data from specific tables in HDR. The user role determines the tables that can be extracted from HDR by the CDW user.

Figure 11 Data Extract Process



The data extract process provides the ability for CDW to extract data from specific HDR tables. Clients requiring HDR data for analytics purposes connect to CDW for executing analytic queries and obtaining reports. A dedicated database account and roles are used on the HDR side to restrict access for CDW to only the view created for presenting the required data. Figure 11 Data Extract Process shows the details of the implementation and user data extraction process. The view is created in HDR to present data for extraction from the appropriate HDR tables. CDW schedules data extraction and storage execution for extracting and storing the data in CDW. Clients that require the data login to CDW and execute queries on CDW tables to meet their data analytics and reporting requirements.

Table 17 Data Extract Process Steps

| Step | Description | Alternate Process |
| --- | --- | --- |
| 1 | Once the HDR database development team receives the requirements for analytics data requirements, the team determines the appropriate SQL to develop and creates a view that can be used for extracting the data. | If the view already exists, no further action is taken by the HDR database development team. |
| 2 | HDR database administrator creates a dedicated database account and roles in HDR for CDW to use to query the above view for extracting data to the CDW. | If the user and user role already exist, no further action is taken. |

| Step | Description | Alternate Process |
| --- | --- | --- |
| 3 | The view, user, roles are tested and deployed to production. |  |
| 4 | CDW dedicated user logs into HDR database and queries the view, performs necessary transforms, and saves the data into the CDW table. |  |
| 5 | The CDW data is queried and compared to data presented in the view in HDR. They should match. |  |
| 6 | Clients requiring the data, login into CDW and run their queries against the CDW table. |  |

#### Data Import Process

Data import illustrated in Figure 12 Data Import Process is available directly on the HDR database. Access will be restricted to specific tables wherein a predetermined CDW user will be granted create and update capabilities to update VSSC rurality data. The user role determines the tables where data can be imported from CDW by the CDW user.

Figure 12 Data Import Process



The data import process provides the ability for CDW to import VSSC rurality data directly to the HDR DB via SQL in support of aggregating HTH Census data by rurality. The HTH reporting requirements for aggregating data by rurality requires regularly updated rurality data. Rurality data is provided by the VSSC group and staged in CDW data stores. When needed, the CDW user will login to the HDR database directly and create or update rurality data in tables allocated for storage of the VSSC rurality data. A dedicated database account and specific roles have been created by the HDR team that restricts access for writing data to the HDR to low transaction timeframes and to only those tables required to be updated as rurality data changes over time. The process for importing rurality data is identified in the table below.

Table 18 Data Import Process Steps

| Step | Description | Alternate Process |
| --- | --- | --- |
| 1 | HDR database development team creates the database table(s) necessary to contain the rurality data based on HTH provided requirements. | If the tables already exists, no further action is taken by the HDR database development team. |
| 2 | HDR database administrator creates a dedicated database account and roles in HDR for CDW to use to create or update rurality data in the HDR. | If the user and user role already exist, no further action is taken. |
| 3 | The tables, user, and roles are tested in integrated test environments against CDW test systems and datasets and promoted to production for use by CDW following regular release processes. | If tables, user, and roles already exist, no action is taken. |
| 4 | CDW dedicated user logs into HDR database and creates or updates rurality data based on the rurality data staged in the CDW data store. | If no updates to rurality data are present, no action is taken. |
| 5 | ARS utilizes VSSC provided rurality data in support of data aggregation and reporting. |  |

#### Project Conceptual Data Model – FtP

The HDR 3.x data model consists of entities that are patterned after the enterprise logical data model (ELDM). Data domains represent subject areas and there are no relationships maintained in the model between subject areas. A majority of the data domains are de-normalized and all data is captured in a single entity.

#### Project Conceptual Data Model – HTH

The Aggregate Read Service (ARS) web service returns aggregate data used for reporting on HTH data. ARS utilizes data models developed through collaboration between HTH and HDR. HDR 3.15 provides census write modifications, validation and handling of census activity reports and patient satisfaction survey titles. A sample of the HDR 3.15 data model for reporting is illustrated in Figure 13.

Figure 13 Census Read Data Model for Reporting

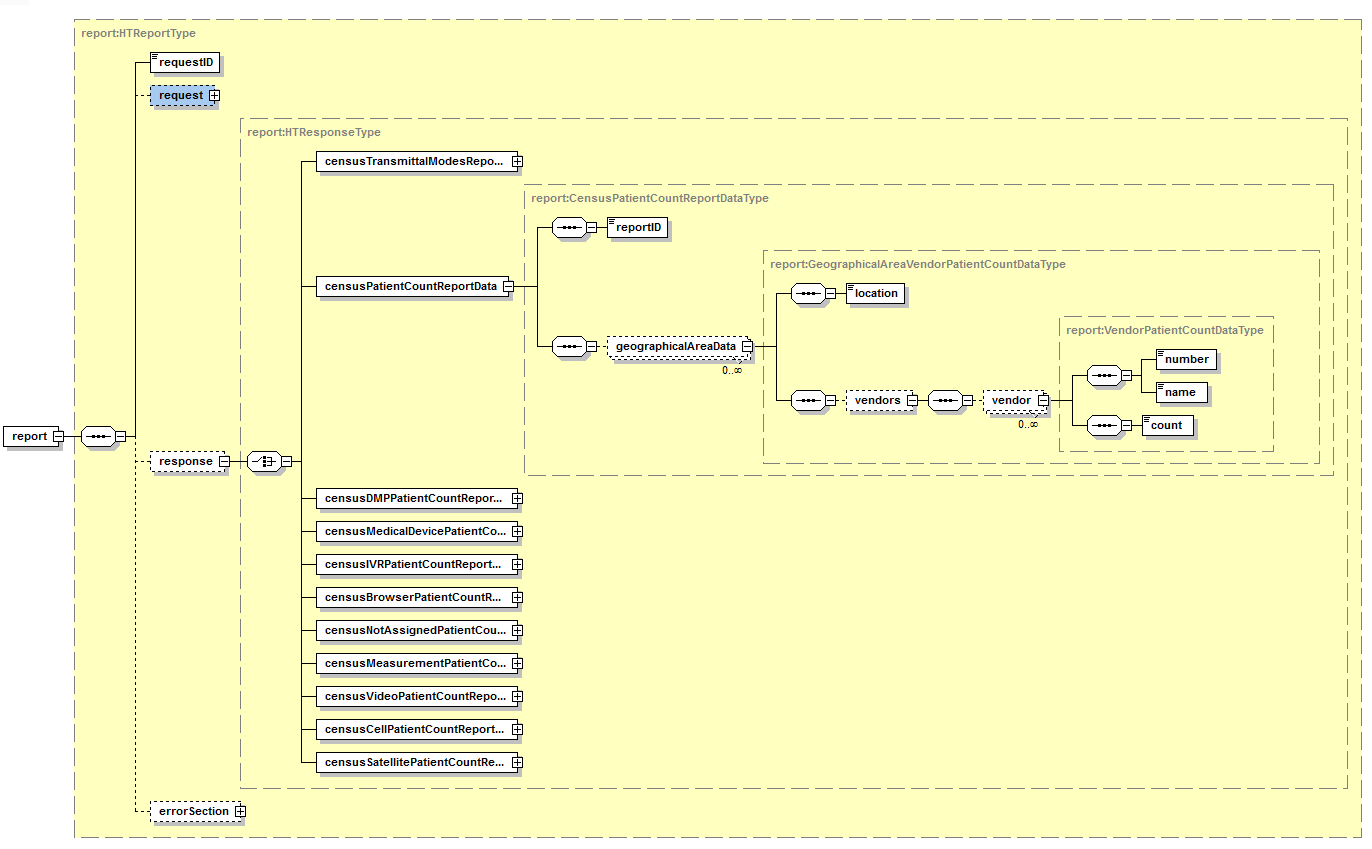


Figure 14 Example of XML report data for the model above

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<report:report xmlns:htFilter="HTReportFilter" xmlns:report="Report" xmlns:reportFilter="ReportFilter">

            <requestID>1</requestID>

            <request>

                        <censusPatientCountReportFilter>

                                    <reportID>CENSUS\_TOTAL\_REPORT</reportID>

                                    <startDate>2014-04-27</startDate>

                                    <endDate>2014-05-03</endDate>

                                    <geographicalArea>

                                                <location>All VISNs</location>

                                    </geographicalArea>

                                    <vendor>All Vendors</vendor>

                        </censusPatientCountReportFilter>

            </request>

            <response>

                        <censusPatientCountReportData>

                                    <reportID>CENSUS\_TOTAL\_REPORT</reportID>

                                    <geographicalAreaData>

                                                <location>VISN 1</location>

                                                <vendors>

                                                            <vendor>

                                                                        <number>200T4</number>

                                                                        <name>ViTel Net</name>

                                                                        <count>12</count>

                                                            </vendor>

                                                            <vendor>

                                                                        <number>200T6</number>

                                                                        <name>ATI</name>

                                                                        <count>11</count>

                                                            </vendor>

                                                            <vendor>

                                                                        <number>200T7</number>

                                                                        <name>Telehealth Cardiocom</name>

                                                                        <count>11</count>

                                                            </vendor>

                                                </vendors>

                                    </geographicalAreaData>

                                    <geographicalAreaData>

                                                <location>VISN 2</location>

                                                <vendors>

                                                            <vendor>

                                                                        <number>200T6</number>

                                                                        <name>ATI</name>

                                                                        <count>1</count>

                                                            </vendor>

                                                </vendors>

                                    </geographicalAreaData>

                                    <geographicalAreaData>

                                                <location>VISN 21</location>

                                                <vendors>

                                                            <vendor>

                                                                        <number>200T6</number>

                                                                        <name>ATI</name>

                                                                        <count>5</count>

                                                            </vendor>

                                                </vendors>

                                    </geographicalAreaData>

                                    <geographicalAreaData>

                                                <location>VISN 23</location>

                                                <vendors>

                                                            <vendor>

                                                                        <number>200T1</number>

                                                                        <name>Telehealth Health Hero</name>

                                                                        <count>1</count>

                                                            </vendor>

                                                            <vendor>

                                                                        <number>200T4</number>

                                                                        <name>ViTel Net</name>

                                                                        <count>1</count>

                                                            </vendor>

                                                            <vendor>

                                                                        <number>200T5</number>

                                                                        <name>Viterion</name>

                                                                        <count>1</count>

                                                            </vendor>

                                                            <vendor>

                                                                        <number>200T6</number>

                                                                        <name>ATI</name>

                                                                        <count>7</count>

                                                            </vendor>

                                                            <vendor>

                                                                        <number>200T7</number>

                                                                        <name>Telehealth Cardiocom</name>

                                                                        <count>2</count>

                                                            </vendor>

                                                            <vendor>

                                                                        <number>200T8</number>

                                                                        <name>Authentidate</name>

                                                                        <count>2</count>

                                                            </vendor>

                                                </vendors>

                                    </geographicalAreaData>

                        </censusPatientCountReportData>

            </response>

            <errorSection/>

</report:report>

### Database Information

Refer to Section 5 Data Design.

Table 19 Database Inventory

| Database Name | Description | Type | Steward |
| --- | --- | --- | --- |
| N/A |  |  |  |

### User Interface Data Mapping

Not applicable. HDR does not have a user interface.

#### Application Screen Interface

Not applicable.

#### Application Report Interface

Not applicable.

#### Unmapped Data Element

Not applicable.

## Conceptual Infrastructure Design

### System Criticality and High Availability

The HDR 3.x system and associated services are implemented as Java EE EJB that are exported as web services and are deployed in two WebLogic cluster environments. One cluster is configured to access HDR database only and the other is configured to access both HDR DB and various VistA instances through HDRDAT and VistA HL7 Interface. All operations supported by HDR services are stateless to reduce system complexity and resilience to system failure. High availability and scalability are provided through the use of WebLogic clustering technology.

Hardware network load balancers manage connections to the WebLogic managed nodes of each cluster. It can further scale by adding more hardware and managed nodes to the WebLogic cluster. The cluster environment provides the ability to stop and restart one or more managed servers with little to no effect on the processing of requests.

HDR 3.x uses HDRDAT to access VistA data by means of Cache persistent objects that provide SQL access via JDBC connections. HDRDAT is installed on all of the VistA systems and runs on the Caché database server in Engineering Change Proposal (ECP) configurations.

HDR 3.x uses VistA HL7 Interface to support writing data to VistA using HL7 logical link connections that are configured by VistA administrators. VistA HL7 package is installed on all of the VistA systems.

HDR 3.x uses the HDR database which utilizes Oracle Real Application Clusters (RAC) that provide high availability and scalability.

These system high availability requirements are met through the implementation of application and database clusters, as well as Disaster Recovery (DR) servers at Hines Information Technology Center (HITC) to ensure that there is continuity of operations in the case of a disaster. In order to meet RTO/RPO, the database servers in the disaster recovery environment at HITC are updated instantaneously as they are configured as ‘active’ hot stand-by nodes, thus ensuring minimal loss of data.

A stand-by environment has been built at AITC to facilitate service recovery in the event that the primary production environment (not standby) is impacted.

AITC currently offers a RTO/RPO of 12 hour/2 hours. If there is a disaster that results in a total outage at AITC, the backup HDR DR site at HITC will take over operations. Refer to Table 3, PERF 02 availability requirement.

### Special Technology

Not Applicable.

### Technology Locations

The CDS application is supported in the Development, Dev Preview, Performance and Load Testing, SQA, and Production environments. The Development environment is used by CDS developers for unit testing of the CDS application. Once tested here, the CDS application is deployed in the SQA and Dev Preview environments for client integration testing. The Developer systems in Salt Lake City (SLC) is comprised of the developer’s laptop or Linux-based virtual desktop connected to WebLogic servers and database servers that are installed in a Linux environment. The virtual desktop environment is located in SLC along with the CDS xx deployed environment and VistA test accounts that are utilized by the CDS application. It is a non-cluster WebLogic environment in which the CDS application is deployed. The Development environment currently uses two VistA shadow instances. The virtualized desktops specifications are as follows:

* 2 CPU’s
* 4 GB Memory
* 40 GB HDD space

Table 20 Technology Location Details

| Technology Component | Location |
| --- | --- |
| Development VDT | SLC |
| Workstations | SLC |
| Special Hardware | SLC |
| VistA Shadow Test Accounts | SLC |
| HDR Database | SLC |
| Application Servers | SLC |
| Development Laptop | SLC/Remote |
| Workstations/Laptop | SLC/Remote |
| Special Hardware | N/A |
| VistA Shadow Test Accounts | SLC |
| HDR Database | Developer Laptop |
| Application Servers | Developer Laptop |

After the Dev Preview, CDS is deployed in the SQA environment. This environment is used for system integration testing including the regression testing of CDS and its associated services/components. It is a WebLogic cluster environment in which the CDS application is deployed. This environment uses two copies of seven VistA accounts in two separate SQA environments. The WebLogic cluster environment contains one cluster with four managed nodes. The database servers utilize Oracle RAC software. This environment currently uses three VistA shadow instances.

### Conceptual Infrastructure Diagram

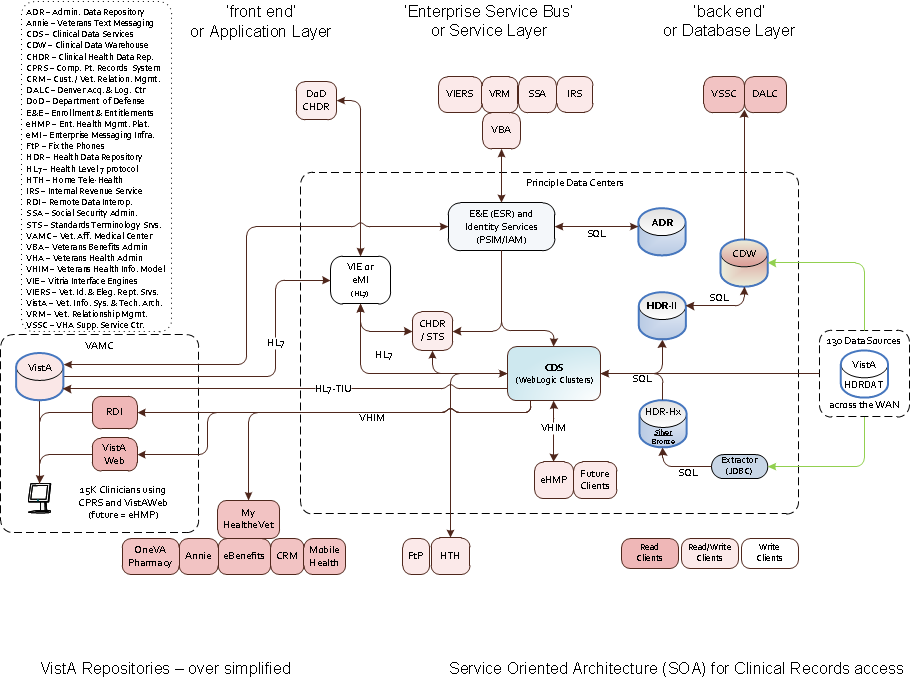
#### Location of Environments and External interfaces

Refer to Section 3.3.3 Technology Locations.

#### Conceptual Production String Diagram

Figure 15 Production String Environment illustrates the detailed production deployment environment showing an instance of the HDR 3.x system deployed to a WebLogic managed node within a cluster, and the cluster’s relationship with the Oracle RAC and Cache cluster configurations.

Figure 15 Production String Environment

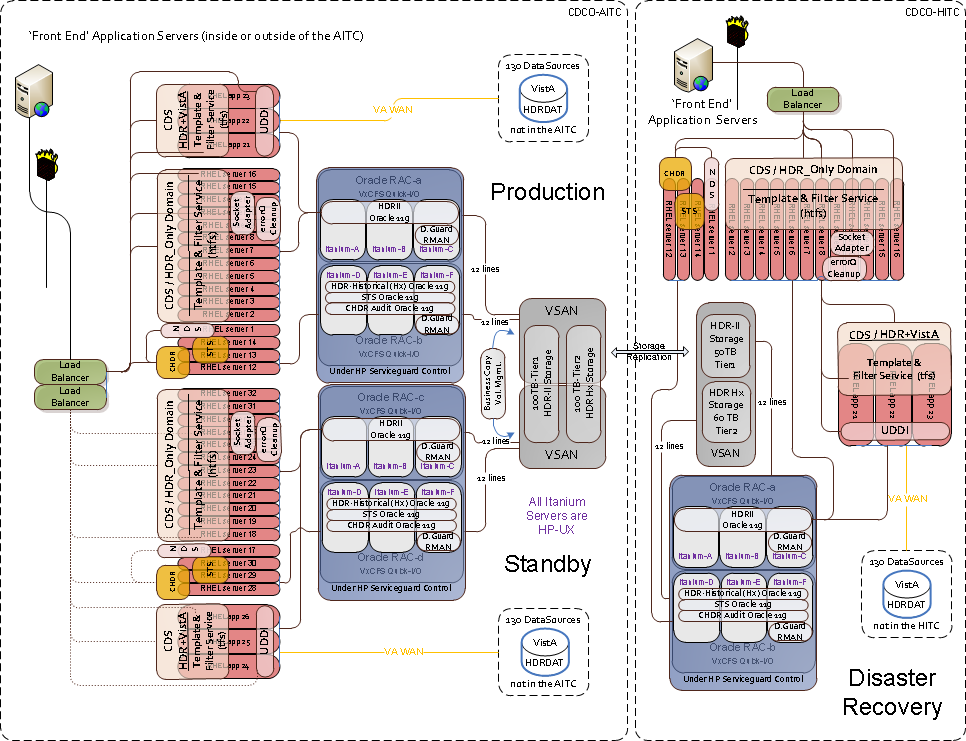


# System Architecture

## Hardware Architecture

Figure 16 HDR General Support System (GSS) illustrates the hardware details.

Figure 16 General Support System Diagram



## Software Architecture

The complete descriptions of the software architecture of the HDR 3.x system and its subsystems for data requests and responses are detailed in Volume 2 of this SDD. In this release, HDR DB and the HDR service interfaces will be modified to support client requirements as needed. All other architecture components do not vary from the previous release of the system..

The architecture improves the performance of specific operations, in particular operations that rely only on HDR database to satisfy all data requirements. The WebLogic cluster is configured with a single data source – HDR DB. This allows the services deployed in the cluster to access data only from HDR DB and thus isolates them from VistA connection-related issues. All Write operations to store data in HDR Database from all clients, except CHDR make use of the services in the HDR-Only Cluster. In addition RDI and HTH use HDR-Only services to retrieve data from HDR.

The existing WebLogic cluster that is configured to access both HDR DB and VistA continues to be operational and is used to support operations that obtain data from both HDR DB and VistA. CHDR is the only client that continues to use this cluster for its Write operations to HDR database. JMS Queues that accept asynchronous write requests for storing data in VistA through FPDS are configured on this domain and are used by FtP for creating and updating notes in VistA. All current CDS 3.x read clients, except HTH are served by the services deployed in this cluster.

All JMS Queues that accept asynchronous Write requests for storing data in HDR Database are configured on the HDR-Only WebLogic cluster. ARS, which provides HTH reports to HTH, is deployed in the HDR-Only cluster. CDS, FPDS and TAFS are deployed in both the WebLogic clusters and clients access the appropriate load balancer end points.

## Network Architecture

Figure 15 illustrates the high level position of a Service Oriented Architecture (SOA) in the delivery of data to Repositories customers / clients. Repositories clients are not end users but applications that the clinicians (end users) depend upon in delivering care to the Veterans. Centralized data centers are required as the clinicians that use our clients’ products are located across the entire VA wide area network, including Puerto Rico and Manila, Philippines).

This illustration shows the RDI and VistA Web clients at the VA Medical Centers. This is because these applications are specifically for clinical use but they may actually reside in other locations.  RDI uses the load balancer endpoint that connects to the HDR-Only WebLogic cluster.

HDR system is deployed across two WebLogic domains, HDR-Only and HDR + VistA. Clients that store and retrieve data from HDR database only use the HDR-Only WebLogic domain endpoints while clients that store and retrieve data from both HDR Database and VistA use the HDR + VistA WebLogic domain endpoints.

All communication to the SOAP and REST web services deployed in the HDR-Only WebLogic domain use VIM supported through XML payloads. JMS interfaces that are exposed on the HDR-Only WebLogic domain support HL7 payload that is transformed to VIM by the Message Mediator service prior to processing. New clients that request access to the HDR services on the HDR-only WebLogic domain are supported using VIM models.

VistA is a NoSQL database developed in MUMPS and deployed on Intersystems Cache. HDRDAT is a layer developed by HDR that provides an interface to all VistA data and VistA APIs through SQL and is deployed at all VistA sites. The HDRDAT interface is used by HDR to access VistA data. This access is isolated to the WebLogic cluster that is configured with both HDR and VistA data source detail. This configuration thus provides SOA access to VistA data.

VistA HL7 Interface provides the ability to access VistA using HL7 and is available at all VistA sites. The VistA HL7 interface requires logical links to be configured with sender and receiver information as well as VistA package information in order to process requests. The VistA HL7 interface is used by HDR to create and update notes in VistA, HDR provides a broker service to enable access to all the VistA systems through the VistA HL7 interface. HDR exposes a JMS interface that processes HL7 and re-routes write requests from the client to the VistA HL7 interface and responses from the VistA HL7 interface to the client.

CDW logs into the HDR DB and extracts HTH census and survey data and stores it in the CDW databases and imports VSSC rurality data into the HDR DB using SQL. Extracted data from HDR is used by VSSC and DALC for analytic purposes. VSSC rurality data is used by ARS in support of HTH requirements for reporting on census activity by rurality. HDR restricts access to CDW to specific tables. Refer to Section 3.2.1.10 Data Extract Process.

### CDS and Associated Services

The HDR CDS has three environments available for Production use. The Standby and Disaster Recovery environments are copies of the Production database. Due to the fact that these database copies are in a down state (no connections during replication), there is no active duplication of the CDS applications. All three environments are located in what we call the GSS or Generally Serviceable Systems (synonymous with production).

Figure 16 (GSS Design Diagram) illustrates that the CHDR and Standards and Terminology Service (STS) programs use HDR hardware even though they are separate development teams.

The red Linux servers function as a cluster because of the design of the applications (many are on WebLogic). The blue systems are put together as Serviceguard clusters in HP-UX. On top of these hardware clusters is the Oracle Real Application Clusters (RAC’s). The RAC-a uses tier 1 storage and the RAC-b uses tier 2 storage. These storage units (on a fibre channel Storage Area Network – FCSAN) have the capability of making rapid copies of the volumes (called Business Copy) at the AITC. If at any time the Standby copy gets too far out of sync, the storage unit can make a rapid snap of the Production volumes and they can be used as a new starting point for replication. There is also storage replication going on between the AITC and the HITC. The Disaster Recover copies of the databases are bit for bit copies of the Production databases (just not turned on).

## Service Oriented Architecture (SOA) / ESS

Repositories program architecture is an SOA that enables maintaining logical divisions in a complex system. It is designed and implemented using modular component-based principles that promote loose coupling. Modules contain their own discreet and encapsulated purpose and interact with one another on a contract basis and can be maintained and modified independently of one another as long as the contract remains unchanged. HDR 3.15 VIP continues this modular SOA architecture to realize the requirements outlined in section 2.3, when applicable. Figure 17 illustrates the architecture layers, modules and components, along with the external services accessed by the Repositories components.

Figure 17 Repositories SOA Architecture



## Enterprise Architecture

At its core, HDR 3.x system is a Java EE framework that takes advantage of Java Spring and dependency injection to extend and implement parallel components required to support the differing needs of its clients. The features the system provides can be accessed by its clients through various means as suited to the client, such as synchronous SOAP or REST service calls, asynchronous messaging through JMS, and TCP/IP connections over sockets. The Java EE framework interacts with the various data sources using JDBC, Hibernate and TCP/IP. Internal support for logging, auditing and error handling are implemented as services. Patient identity correlation, primarily used for identifying the VistA locations at which the patient was seen, is achieved using SOAP calls to the external Identity Management Service.

The HDR 3.x system adheres to the VA Enterprise Architecture (EA) and makes use of approved tools, standards and rules as identified by the VA Technical Reference Model (TRM) / Standards Profile (SP) located at http://DNS .DNS. By using these standards, the programs take advantage of the current technologies that are adapted by VA and promotes interoperability, portability and adaptability within systems and promotes quality assurance. In addition to using SOA principles as defined by the VA EA team, HDR 3.x uses TRM approved tools such as WebLogic 12.x and Oracle 11g as its application server and database server respectively.

The HDR 3.x team continuously monitors TRM and upgrades the software, tools and libraries that are used to build the system on a regular basis to ensure compliance with VA EA. Table 21 describes the technologies used by HDR and their status.

Table 21 HDR TRM Technologies / Tools

|  | **Tool Name** | **Version** | **Purpose** | **Comments/Actions** | **Approved** | **Approved**  **w/Constraints** | **Version Upgrade**  **Required** | **Estimated Completion Date** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Another Tool for Language Recognition (ANTLR) | 4.1 | Language Parser for Java | Version 4.1 Approved w/Constraints thru Q4 CY16, then Divest thru Q4 CY17. Version 4.5.x Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 2 | Apache Commons BeanUtils | 1.9.1 | Common Utility Classes | Version 1.9.x Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 3 | Apache Commons Codec | 1.10 | Common Utility Classes | Version 1.10 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 4 | Apache Commons Collections | 4.0 | Common Utility Classes | Version 4.0 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 5 | Apache Commons Database connection pooling (DBCP) | 2.1 | Common Utility Classes | Version 2.1 Approved w/Constraints thru Q1 CY17, then Divest thru Q4 CY17. Version 2.1.x Approved w/Constraints thru Q4 CY17. |  | X |  |  |
| 6 | Apache Commons Digester | 3.2 | Common Utility Classes | Version 3.2 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 7 | Apache Commons Jxpath | 1.3 | Java-based implementation of XPath | Version 1.3 Approved w/Constraints thru Q4 CY18 |  | X |  |  |
| 8 | Apache Commons Lang | 3.4 | Common Utility Classes | Version 3.4 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 9 | Apache Commons Logging | 1.2 | Common Utility Classes | Version 1.2 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 10 | Apache Commons Pool | 2.3 | Common Utility Classes | Version 2.3 Approved w/Constraints thru Q1 CY17, then Divest thru Q4 CY17. Version 2.4.x Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 11 | Apache CXF | 3.1.3 | Web Service Framework for Java | Version 3.1.3 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 12 | Apache MINA | 2.0.9 | Multipurpose Infrastructure for Network Application | Version 2.0.x Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 13 | Apache Scout | 1.2.7 | Implementation of JAXR API | Version 1.2.7 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 14 | Aspect-Oriented Programming (AOP) Alliance | 1.0 | Aspect Oriented Programming library | Version 1.0 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 15 | Attachmate Reflection X Advantage | 5.0 | X server for running X client applications with enhanced features | Version 5.x Approved thru Q4 CY16, then Divest thru Q4 CY17. Version 5.1 Approved thru Q4 CY17 | X |  |  |  |
| 16 | Cache | 2014.1.3 | JDBC Driver for Connecting to IS Cache | Version 2014.1.3 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 17 | dom4j | 1.6.1, 1.6.1-PATCH01 | Flexible XML Framework for Java | Version 1.6.1 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 18 | Easymock | 3.3 | Library for Generating Mock Test Objects | Version 3.3 Approved thru Q4 CY17.  9/22/16 - A request to add version 3.4 was submitted to TRM.  9/23/16 - TRM tentatively Approved w/Constraints thru Q4 CY17 for version 3.4. |  | X |  |  |
| 19 | e(Core) | 2.10 | Eclipse EMF Support | Version 2.10 Approved w/Constraints thru Q4 CY16, then Divest thru Q4 CY17. Version 2.11.x Approved w/Constraints thru Q4 CY17.  E(Core) is a component under Eclipse Modeling Framework (EMF) |  | X |  |  |
| 20 | Eclipse Classic | 4.5 | Integrated Development Environment | Version 4.5 Approved w/Constraints thru Q4 CY17. |  | X |  |  |
| 21 | Ehcache | 3.1.2 | Ehcache is a Java distributed cache for general purpose caching, Java Enterprise Edition and lightweight containers. | Versions 3.0.x, 2.10.x, and 4.3.x Approved w/Constraints thru Q4 CY17.  A request to update Ehcache entry with version 3.1.2 was submitted 10/4/16, awaiting TRM reply. |  | X |  |  |
| 22 | Hermes JMS | 1.14 | Java Messaging Service API | Version 1.14 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 23 | Hibernate Object/Relational Mapping (ORM) | 4.3.10Final | Object Relational Mapping Library | Version 4.3 Approved w/Constraints thru Q4 CY16, then Divest thru Q4 CY17. Version 5.1.x Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 24 | HL7 Application Programming Interface (HAPI) | 2.2 | HL7 API for Java | Version 2.2 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 25 | HP-UX | 11iv3 | HP-UX (Hewlett Packard Unix) is Hewlett-Packard`s proprietary implementation of the Unix operating system | HDR has an approved AERB Waiver to use this Unapproved TRM technology | X |  |  |  |
| 26 | Java SE | 7 (aka 1.7) | Java Development | Version 7 Divest thru Q4 CY16, then Unapproved. Version 8u92 Approved w/Constraints thru Q4 CY17.  Plan: Upgrade to version 8u92 |  | X | X | 11/30/16 |
| 27 | JavaScript Object Notation Data Interchange Standard (JSON) Path | 2.1 | Provides a simple way to extract parts of a given document using XPath like notation. | Version 2.1 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 28 | JavaScript Object Notation Data Interchange Standard (JSON) Smart | 2.2 | A performance focused, JSON processor library | Version 2.2 Approved w/Constraints thru Q4 CY2017 |  | X |  |  |
| 29 | Jenkins Continuous Integration Server | 1.651.1 | Continuous Integration | Version 1.6x Approved w/Constraints thru Q2 CY17, then Divest thru Q4 CY17. Version 2.x Approved w/Constraints thru Q4 CY17. |  | X |  |  |
| 30 | JetBrains Annotations | 9.0 | Java annotations support library. | Version 9.0 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 31 | JSON in Java | 20151123 | JSON is a light-weight, language independent, data interchange format. This library provides a Java implementation of the JSON standard | Version 20151123 Approved w/Constraints thru Q4 CY2018 |  | X |  |  |
| 32 | Junit | 4.12 | Java unit test development and testing framework | Version 4.12 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 33 | LinkChecker | 9.3 | LinkChecker is a free, GPL licensed website validator. LinkChecker checks links in web documents or full websites. | Version 9.3 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 34 | Log4J | 2.4 | Logging Utility for Java | Version 2.4 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 35 | Maven | 3.2.3 | Build Automation & Dependency Management | Version 3.2.3 Approved w/Constraints thru Q2 CY17, then Divest thru Q4 CY 17. Version 3.3.x Approved w/Constraints thru Q4 CY17. |  | X |  |  |
| 36 | Messaging Workbench (MWB) | 6.8 | Multipurpose productivity software for Health Level Seven (HL7) V2 implementers | Version 6.8 Approved thru Q4 CY16, then Divest thru Q4 CY17. Version 6.8.x Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 37 | Mockito | 1.10.19 | Mock Testing Framework | Version 1.10.x  Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 38 | Nagios Core | 4.0.x | Open source monitoring application for system resource metrics, network protocols, applications, servers, and network infrastructure | Version 4.0.x Approved w/Constraints thru Q4 CY17. |  | X |  |  |
| 39 | Nexus | 2.11.4-01 | Infrastructure and services for organizations that use repository managers to obtain and deliver software | Version 2.11 Approved w/Constraints thru Q4 CY16, then Divest thru Q4 CY17. Version 3.0 Approved w/Constraints Q4 CY17 |  | X |  |  |
| 40 | Notepad++ | 6.x | Text editor for Windows used for source code editing | Version 6.x Approved w/Constraints thru Q1 CY17, then Divest thru Q4 CY17 |  | X |  |  |
| 41 | Oracle Database | 11.2.0.4 | Relational Database Management System | Version 11.2.x Approved w/Constraints thru Q1 CY17, then Divest thru Q4 CY17; version 12.1.x Approved/Constraints thru Q4 CY17 |  | X |  |  |
| 42 | Oracle Enterprise Manager (OEM) | 11.1.0.1 | Database Management | Version 11.1.0.1 Approved w/Constraints thru Q1 CY16, then Divest thru Q4 CY16, then Unapproved Q1 CY17. Version 13.1.0.0 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 43 | Powermock | 1.6.4 | Java Class Mocking Framework | Version 1.6.x Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 44 | Putty-CAC | 0.66 | Putty-CAC is Windows terminal emulation software that supports the Secure Shell (SSH) protocol to access remote systems. It is a modified version of Putty SC and it supports Smartcard authentication | Version 0.66 is Approved w/Constraints thru Q4 CY17. |  | X |  |  |
| 45 | Rational DOORS Next Generation (RDNG)  (formerly: Rational Requirements Composer) | 6.0.1 | Requirements management platform that enables organizations to create or enhance requirements processes and provides visualization and collaboration capabilities. | Version 6.0.x Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 46 | Rational Quality Manager | 6.0.1 | Web-based centralized test management environment, SQA Software | Version 6.0.x Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 47 | Rational Team Concert | 6.0.1 | Software source control, workspace management, and supports parallel development | Version 6.0.x Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 48 | Red Hat Enterprise Linux (aka: RHEL) | 5.11 | Linux version of Red Hat that provides installation routines, pre-configured components, and support services that make the installation-maintenance of the operating system more robust | Version 5.x Approved w/Constraints thru Q4 CY15, then Divest thru Q4 CY16, Unapproved Q1 CY17. Version 6.x Approved w/Constraints thru Q4 CY17.  Plan: Upgrade to version 6.x |  | X | X | 12/31/16 |
| 49 | Red Hat Enterprise Virtualization (Servers)  (aka: RHEV) | 3.5.4 | Server virtualization management solution | Version 3.5.x Approved w/Constraints thru Q4 CY16, then Divest thru Q4 CY17.  Version 3.6.x Approved w/Constraints thru Q4 CY16, then Divest thru Q4 CY17.  No new version to upgrade to listed in TRM |  | X |  |  |
| 50 | Remote Server Administration Tools/Windows AdminPack | Windows 7 SP1 | Remote Desktop | Version 7 SP1 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 51 | Saxon | 9.7.0-7 | XSLT engine and Xquery processor | Version 9.7 Approved w/Constraints thru Q4 CY17. |  | X |  |  |
| 52 | SLF4J | 1.7.x | Simple Logging Façade for Java | Version 1.7.x, Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 53 | SoapUI | 5.0.0 and 5.1 | Open source web service testing application for service-oriented architectures (SOA) | Versions 5.0 and 5.1 Approved w/Constraints thru Q1 CY17. Version 5.2.x Approved w/Constraints thru Q4 CY17. |  | X |  |  |
| 54 | Spring Framework | 4.2.7 .RELEASE | Comprehensive infrastructure supporting the development of Java Applications | Version 4.2.x Approved w/Constraints thru Q4 CY17  9/15/16, Submitted update to TRM to add version 4.3.x which works with Hibernate ORM v5.1.x |  | X |  |  |
| 55 | TOAD Development Suite for Oracle | 12.8.0.49 | Database Development | Version 12.8 Approved w/Constraints thru Q4 CY18. |  | X |  |  |
| 56 | Vagrant | 1.x | Vagrant is an open-source solution for automating the creation and configuration of consistent, portable and distributable virtual development environments and their specific technical requirements. | Version 1.x Approved w/Constraints thru Q4 CY17. |  | X |  |  |
| 57 | VirtualBox | 5.0.x | Oracle VM VirtualBox is a virtualization software package, a type 2 hypervisor, i.e. it operates as an application on top of an existing operating system. This host application allows additional guest operating systems to be loaded and run each with its own isolated virtual environment. | Version 5.0.x is Approved w/Constraints thru Q4 CY17. |  | X |  |  |
| 58 | WebLogic Server | 10.3.5, 12.1.2 | J2EE Application Server | Version 10.3.5 Approved/w Constraints thru Q4 CY16.  Version 12.1.x Approved w/Constraints thru Q3 CY17  Plan: WebLogic version 10.3.5 will be removed from the HDR TRM list prior to 12/31/16; will be replaced with the eMI WebSphere Service Registry and Repository (WSRR) solution |  | X |  |  |
| 59 | Windows Server | 2008 R2 SP1, 2012 | SharePoint and other support application servers | Versions 2008 R2 SP1 and 2012 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 60 | WinSCP | 5.7.x | FTP / SFTP Client for Windows | Version 5.7.x Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 61 | WinZip | 20.0 | Software for file compression and archiving | Version 20.0.x Approved w/Constraints thru Q4 CY18 |  | X |  |  |
| 62 | Xerces2 Java | 2.11 | Library for parsing, validating and manipulating XML documents | Version 2.11 Approved w/Constraints thru Q4 CY17 |  | X |  |  |
| 63 | XMLSpy | 2015 | Markup language editor and integrated development environment for creating, modifying, transforming, and debugging XML-related technologies | Release 2015 Approved thru Q1 CY17, then Divest thru Q4 CY17. Version 2016r2 Approved w/Constraints thru Q4 CY17. |  | X |  |  |
| 64 | XMLUnit for Java | 1.5 | Support Library for Testing Java Classes with XML | Version 1.5 Approved thru Q1 CY17, then Divest thru Q4 CY17. Version 2.1.x Approved w/Constraints thru Q4 CY17. | X |  |  |  |

Table updated 10/12/2016

# Data Design

## DBMS Files

No changes needed for FtP client.

The following changes will support the HTH client for the HDR 3.15 VIP Build 1 release:

1. Census – Transformation Application Messaging (RRC# 796063)
2. New database schema for installing the Quartz Enterprise Job Scheduler.
3. New Quartz job to monitor completed, but unacknowledged transforms.
4. New columns in table SURVEY.Census\_Complete\_Week to track the following data:
   * Acknowledgment status (NEW, READY, INPROGRESS, SENT)
   * Timestamp of status change
   * Request ID (HL7 messaging control ID)
   * Receiving application
   * Receiving facility
   * Sending application
   * Station number
5. New table to log transformation errors for each ingest report.
6. New parameter data to define how many transform errors can occur before discontinuing further data validation for each ingest report.
7. New stored procedure to determine status of ingest transformations and create required data for returning the application message (AA or AR).
8. Update existing process code to not to send an application message for each census segment received.
9. Census - Transmission Application Messaging (RRC#799956)
10. Update existing process code to send one application reject (AR) message upon encountering the first ingest segment in error. Thereafter, no more AR messages will be sent for the ingest. Individual segment validation will continue until all received segments of the ingest have been validated.
11. Census segment reception time monitoring (RRC#612872)
12. New quartz job to check for incomplete census ingests that have exceeded the predefined timeout limit, and send Application Reject (AR) message for expired censuses.
13. New stored procedure determine if ingest transmission has expired before receiving all ingest segments.
14. Uses new ACK status column from requirement 1 to determine if message needs to be sent.
15. Validation of the total number of Census Report Segments (RRC#799954)
16. Update current stored procedure SURVEY.Census\_Master\_Record\_Mgt.Census\_Master\_Records to check for too many ingests segments being sent and send one AR message.
17. 2.0 Patient Satisfaction Survey Title (RRC#716736)
18. Update report title in table SURVEY.Survey from ‘Patient Satisfaction’ to ‘VA Home Telehealth Patient Satisfaction Survey’ for version 2.0
19. Update stored procedure SURVEY.Ht\_Survey\_Report.Patient\_Satisfaction\_Data to use new title.

## Non-DBMS Files

Not Applicable.

## Data View

The HDR system serves as the data service layer and does not present data directly to the end user.

# Detailed Design

The Hardware and Software Detailed Design subsections of Section 6 are included in HDR 3.15 SDD, Volume 2. Volume 2 describes the major components of the HDR 3.x system, including a technical design overview, module design, processing, data structure, configurations, process flow diagrams and other diagrams as needed. HDR provides back end processing only and does not include user interfaces or interactions.

The following Subsections are described in detail in the *HDR 3.15 VIP Build 1 System Design Document, Volume 2*.

## Hardware Detailed Design

Refer to Figure 16 HDR GSS in this volume and in Volume 2 of the HDR 3.15 VIP Build 1 SDD.

## Software Detailed Design

Refer to Volume 2 of the HDR 3.15 VIP Build 1 SDD.

1. Volume 2 contains detail technical design. In the event there are modifications to any of the detailed design, it will be indicated to the reviewer.

## Network Detailed Design

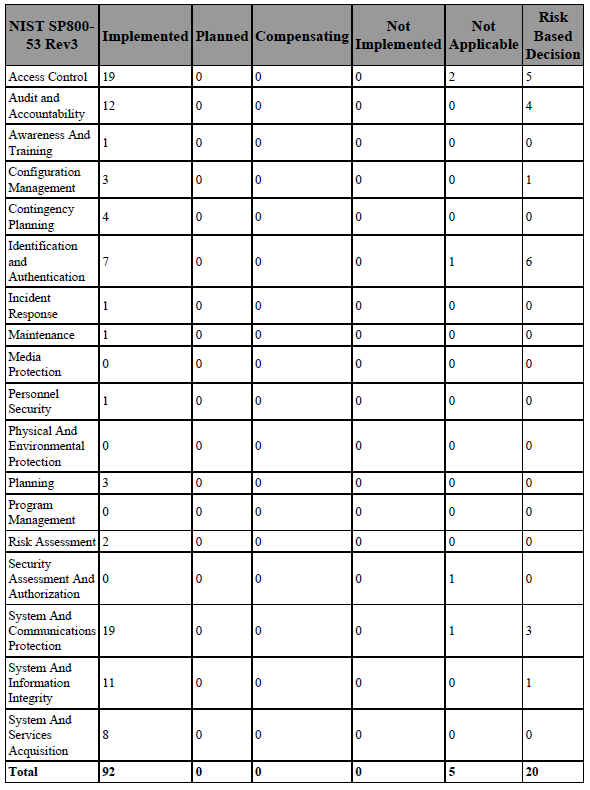
Refer to previous diagrams in Section 4.3 Network Architecture.

## Security and Privacy

### Security

HDR is a major application and is classified at FISMA HIGH security category as it stores PII and PHI. HDR SCA was performed against the System Security Plan (SSP) and all supporting documentation for HDR. The SSP was created based on NIST 800-53 rev. 3 security controls consisting of 18 controls, plus the 8 privacy controls. The SSP discusses the management, operation and technical requirements and shows details and status each of the controls. The following table summarizes the security control status summary:

Table 22 Security Control Status



All of the accreditation documentation has been completed along with the supporting documents for the SSP. For access to the SSP, PTA and other accreditation and supporting documents, please contact the HDR ISO.

### Privacy

HDR is a privacy sensitive IT system. A Privacy Threshold Analysis (PTA) was performed and approved in April 2014. A Privacy Impact Assessment (PIA) was completed in August of 2011 and is publicly available at http://www.privacy.DNS /privacy\_impact\_assessment.asp. There have been no major changes since that time and thus the privacy officers have determined that the PTA performed in April is sufficient. The eight (8) Privacy control families from NIST 800-53 rev. 4 have been documented in the SSP:

* Authority and Purpose (AP)
* Accountability, Audit, and Risk Management (AR)
* Data Quality and Integrity (DI)
* Data Minimization and Retention (DM)
* Individual Participation and Redress (IP)
* Security (SE)
* Transparency (TR)
* Use Limitation (UL)

All of the accreditation documentation has been completed along with the supporting documents for the SSP. For access to the SSP, PTA and other accreditation and supporting documents, please contact the HDR ISO.

## Service Oriented Architecture / ESS Detailed Design

See section 4.4 for an overview of the HDR 3.x SOA/ESS design. Data exchanged by HDR 3.x services and other interfaces is computable data and is primarily Patient Health Information (PHI). Other types of information such as survey results are also exchanged and stored in HDR. HL7 and VistA terminology standards apply to the information exchanged by the HDR 3.x clients with the HDR 3.x services and interfaces.

The following sub-sections delineate the clients that consume each of the HDR 3.x services and/or interfaces. These sections provide details that can be used by Enterprise Service Registry and Repository (ESRR) for exposing the service and interfaces to the VA enterprise. The individual files and links to the actual services are not included in this table due to security constraints and will be provided directly to ESRR.

### Service Descriptions

#### FPDS JMS Interface

The FPDS JMS interface supports asynchronous writes.

Table 23 FPDS JMS Write Clients

|  |  |  |  |
| --- | --- | --- | --- |
| **HDR 3.x Client System** | **Data Domains** | **Information Exchange Description** | **Information Element (Payload)** |
| FtP | Notes | ​The FtP client places an HL7 payload consisting of TIU Notes data onto a JMS queue that is processed by the FPDS Message Driven Bean (MDB). See FtP specifications for details on the HL7 structure. | Per FtP  Message Exchange Standard: Notes  Payload Format: HL7 |

#### FPDS REST Web Service Read Clients

FPDS provides a REST interface to be used by clients for retrieving data. This aspect of the FPDS service is supported in the existing 3.x WebLogic cluster that provides access to both HDR DB and VistA data sources. The information exchange details for the interaction are as specified in the following table:

Table 24 FPDS REST Web Service Read Clients

| HDR 3.x Client System | Data Domains | Information Exchange Description | Information Element (Payload) |
| --- | --- | --- | --- |
| Health Management Platform (eHMP) | Multiple per VPR  See VistA VPR 1.2 API for details on supported domains. | ​eHMP uses FPDS REST interface to request VPR data from VistA.  The path parameters and the query parameters are used to construct an XML filter as required by the CDS framework. The XML response from CDS that conforms to the template which includes JSON data from VistA VPR and XML wrapper per CDS is returned to FPDS. Filter supports few filter parameters which are supported by default by VPR. If the client requested XML format, the response is transformed to valid XML in FPDS using XSLT prior to returning to the client. If the client requested JSON response, the JSON response is returned by FPDS. The filter and template are internal to CDS. The client needs to pass the names of them in the request. | Template: GenericObservationRead1  Filter: GENERIC\_VISTA\_LIST\_DATA\_FILTER  Payload Format: JSON/XML  VistA API Used: VPR 1.2 JSON |
| FtP | Multiple as per FtP RSDs.  See VistA VPR 1.2 API for details on supported domains. | FtP uses FPDS REST interface to request data from VistA.  The path parameters and the query parameters are used to construct an XML filter as required by the CDS framework. The XML response from CDS that conforms to the template which includes JSON data from VistA VPR and XML wrapper per CDS, is returned to FPDS. Filter supports few filter parameters which are supported by default by VPR. The additional optional domain specific filter parameters are supported by CDS. The additional filter parameter names that begin with ‘@.’ are included as JSONPath elements and applied to the VistA VPR responses prior to being packaged into a response that conforms to the template which includes JSON data from VistA VPR and XML wrapper per CDS. This response is transformed to valid XML in FPDS using XSLT prior to returning to the client. If the client requested JSON response, the JSON response is returned by FPDS. The client needs to pass the names of them in the request. In the case of Radiology, VPR XML is used. | Template: GenericObservationRead1;  Filter: GENERIC\_VISTA\_LIST\_DATA\_FILTER  Payload Formats: JSON and XML  VistA API Used: VPR 1.2 JSON |

#### ARS SOAP Web Service Read Clients

ARS provides a SOAP interface to be used by clients for accessing aggregated data. ARS is deployed in the HDR-Only WebLogic cluster as it provides data from HDR DB only. This interface is used by HTH as shown in the following table:

Table 25 ARS SOAP Web Service Read Clients

| HDR 3.x Client System | Data Domains | Information Exchange Description | Information Element (Payload) |
| --- | --- | --- | --- |
| Home Telehealth (HTH) | * DMPs, * Patient Census * Survey report-related aggregations | HTH uses ARS to obtain aggregated reports on ADL and PS surveys, DMPs and Census data stored in HDR. | Filters:   * HDReportFilter.xsd * HTResponse.xsd * ReportFilter.xsd   Message Exchange Standard: VA ELDM  Payload Format: XSD |

#### CDS REST Web Service Read Clients

CDS provides a REST interface to be used by clients for accessing data. This interface is used by RDI for accessing the VistA data that is cached in HDR. RDI uses the CDS endpoint that is available in the HDR-Only domain for retrieving data. The information exchange details used by RDI are as shown in the following table:

Table 26 CDS REST Web Service Clients

| HDR 3.x Client System | Data Domains | Information Exchange Description | Information Element (Payload) |
| --- | --- | --- | --- |
| Remote Data Interoperability (RDI) component of VistA Computerized Patient Record System (VistA CPRS) | * Allergy * Pharmacy | ​CPRS uses readClinicalData1 operation on CDS Web Service through RDI for order checks. - Gets both CHDR and VistA allergies and medication data from HDR database for a patient - Operation parameters include an XML string that specifies data domain and filter attributes along with the response payload schema name. Response is an XML schema payload | Filters:   * RDI\_IC\_RX\_Single\_Patient\_Filter.xsd * RDIIntoleranceConditionPharmacyRead40010   Message Exchange Standard: VA ELDM (VIM)  Payload Format: XSD |

#### CDS SOAP Web Service Read Clients

CDS provides a SOAP interface to be used by clients for storing and accessing data. Clients using this interface access the CDS endpoint that is available in the WebLogic cluster that is configured with both HDR DB and VistA data sources. This interface is used by various clients for obtaining clinical data as shown in the following table:

Table 27 CDS SOAP Web Service Read Clients

| HDR 3.x Client System | Data Domains | Information Exchange Description | Information Element (Payload) |
| --- | --- | --- | --- |
| Clinical / Health Data Repository (CHDR) | * Allergy * Pharmacy | Call to readClinicalData1 operation on CDS Web Service on HTTP/SOAP to get VistA allergies and outpatient pharmacy data. 1 call per domain - varies by the filer and template schema parameters specified and used to create the payloads | Filters:   * CHDR\_Allergy\_Single\_Patient\_Filter.xsd * IntoleranceConditionRead40010 * CHDR\_RX\_Single\_Patient\_Filter.xsd * PharmacyRead40010   Message Exchange Standard: VA ELDM (VIM)  Payload Format: XML |
| My HealtheVet - Personal Health Record - Online Viewing (MHV) | * Allergy * Laboratory * HTH Vitals | ​Service: CDS; Operation: readClinicalData1 - 1 call per patient per data domain - Uses filters to specify query parameters and specifies response template to use on web service operation. MHV uses HDR/CDS via MDWS for following data from HDR database: Vitals (HTH) Lab C/H (VistA) Intolerance condition (CHDR + VistA) | Filters:   * MHV\_Vital\_Single\_Patient\_Filter.xsd * MHVVitalsignsRead40010 * IC\_Single\_Patient\_All\_Data\_Filter.xsd * MHVIntoleranceConditionRead40011 * Lab\_Single\_Patient\_All\_Data\_Filter.xsd * MHVLabRead40011   Message Exchange Standard: VA ELDM (VIM)  Payload Format: XML |
| VistAWeb (VW) | * Allergy * Pharmacy * HTH Vitals | Call readClinicalData1 on CDS web service for HTH Vitals, CHDR Allergies and CHDR Pharmacy Data | Filters:   * RDIAllergiesPharmacyRead40013.xsd * RDI\_ALLERGY\_RX\_SINGLE\_PATIENT\_FILTER.xsd * RX\_Single\_Patient\_All\_Data\_Filter.xsd * VWPharmacyRead40010 * VW\_VITAL\_Single\_Patient\_Filter.xsd * VWVitalsignsRead40010   Message Exchange Standard: VA ELDM (VIM)  Payload Format: XML |
| FtP | * Vitals (HTH and VistA); * Allergy (DoD and VistA); * Pharmacy (DoD and VistA) | Call readClinicalData1 on CDS web service | Filters:   * VITAL\_SINGLE\_PATIENT\_ALL\_DATA\_FILTER, VitalsignsRead40010 * VW\_ALLERGY\_Single\_Patient\_Filter.xsd * VWAllergiesRead40010 * RX\_Single\_Patient\_All\_Data\_Filter.xsd * PharmacyRead40010 * OutpatientRead1   Message Exchange Standard: VA ELDM (VIM)  Payload Format: XML |
| OneVA Pharmacy/eMI | Pharmacy (DoD and VistA) | Call readClinicalData1 on CDS web service | Filter:   * RX\_Single\_Patient\_Filter.xsd * PharmacyRead40010   Message Exchange Standard: VA ELDM (VIM)  Payload Format: XML |

#### Pathways Clients

Pathways provides a SOAP interface to be used by clients for accessing data. Clients using this interface access the Pathways endpoint that is available in the WebLogic cluster that is configured with VistA data sources. This interface is used by various clients for obtaining data as shown in the following table:

Table 28 Pathways SOAP Web Read Service Clients

| HDR 3.x Client System | Data Domains | Information Exchange Description | Information Element (Payload) |
| --- | --- | --- | --- |
| CRM, MHV, FtP and  MobileHealth | Appointments | Call readData on Pathways SOAP web service for patient Appointments data from VistA | Template:   * AppointmentsRead1   Filter:   * Appointments\_Single\_Patient\_Filter.xsd   Message Exchange Standard: VA ELDM (VIM)  Payload Format: XML |
| FtP | Demographics | Call readData on Pathways SOAP web service for patient Demographics data from VistA | Template:   * DemographicsRead1   Filter:   * Demographics\_Single\_Patient\_Filter.xsd   Message Exchange Standard: Name/Value  Payload Format: XML |
| FtP | VistA Alerts | Call readData on Pathways SOAP web service for Patient Alerts list and details | Templates:   * PatientAlertListRead1 * PatientAlertDetailRead1   Filters:   * PATIENTALERT\_SINGLE\_PATIENT\_LIST\_FILTER * PATIENTALERT\_SINGLE\_PATIENT\_DETAIL\_FILTER   Message Exchange Standard: VHIM XML  Payload Format: XML |
| eBenefits | * Exam Request * Exam | Call readData on Pathways SOAP web service for patient Exam Request and Exam 2507 data from VistA | Template:   * RequestsAndExamsRead1 * RequestsAndExamsRead2   Filter:   * Requests\_And\_Exams\_Single\_Patient\_Filter.xsd   Message Exchange Standard: VA ELDM (VIM)  Payload Format: XML |

#### Socket Adapter Read Clients

HDR 3.x provides a Socket Adapter interface that clients can connect to using TCP/IP for storing data in HDR. When the Socket Adapter sees a message, it places the message on the appropriate CDS Message Mediator JMS queue for further processing. HTH and VistA are the primary Socket Adapter clients that send HL7 messages which are then transformed by the CDS Message Mediator component into XML payload for processing and storing to HDR, as shown in the following table:

Table 29 Socket Adapter Read Clients

| HDR 3.x Client System | Data Domains | Information Exchange Description | Information Element (Payload) |
| --- | --- | --- | --- |
| Home Telehealth (HTH) | * Vitals Measurement * Patient Satisfaction Surveys * Veterans Rand 12 Item Health Survey (VR-12) DMPs * Patient Census | HTH Connects to HDR/CDS Socket Adapter Port and sends HL7 data over TCP/IP; receives acknowledgements and CA and CE messages back  1 call per data domain | ​HL7 2.4 messages  ADL and PS surveys, DMPs and Census data are XML strings wrapped in HL7.  Message Exchange Standard: HL7 2.4  Payload Format: HL7 |
| VistA | Lab | Vista connects to the HDR/CDS Socket Adapter to send VistA lab results (Chemistry and Hematology) for storing in the HDR database. | HL7 2.4 messages  Message Exchange Standard: HL7 2.4  Payload Format: HL7 |
| FPDS (socket client) | TIU Notes | FPDS connects to the VistA HL7 interface to send HL7 TIU requests for storing in VistA. | HL7 2.4 messages  Message Exchange Standard: HL7 2.4  Payload Format: HL7 |
| VistA HL7 Interface | * Acknowledgements * Errors | VistA HL7 Interface connects to the socket adapter to send back CR, AA, responses and errors on TIU Notes create/update request | Message Exchange Standard: HL7 2.4  Payload Format: HL7 |

#### CDS Message Mediator Clients

HDR 3.x provides a JMS interface through the CDS Message Mediator component. This interface is used by the CDS Socket Adapter and VIE to process messages. See section 6.2.1.27.6 for further details on the processing of the HL7 messages received on the JMS queues by Message Mediator. CDS Message Mediator is deployed in the HDR-Only WebLogic domain and makes use of the CDS service components that are available in the HDR-Only WebLogic domain cluster. The interface exchange details are as shown in the following Table:

Table 30 CDS Message Mediator Clients

| HDR 3.x Client System | Data Domains | Information Exchange Description | Information Element (Payload) |
| --- | --- | --- | --- |
| VIE | Allergies, OP, Vitals | VIE connects to the HDR/CDS Message Mediator component to send VistA Allergies, OP, and Vitals data for storing in HDR. | HL7 2.4 messages  Message Exchange Standard: HL7 2.4  Payload Format: HL7 |
| Socket Adapter | Vitals Measurement; ​Patient Satisfaction Surveys; Veterans Rand 12 Item Health Survey (VR-12); DMPs; Patient Census | All data that arrives on the socket connection is placed on the appropriate JMS queues for processing by the Message Mediator | HL7 2.4 messages Survey, Census and DMP are XML strings wrapped in HL7  Message Exchange Standard: HL7 2.4  Payload Format: HL7 |

#### VIM-formatted Write Clients

CHDR sends VIM-compliant allergy and outpatient pharmacy data over a web service for active dual consumers to CDS to be stored in the HDR DB. Conversion from HL7 messages to VIM are managed by CHDR before writing via CDS to HDR DB, bypassing the JMS queues and Message Mediator.

The interface exchange details are as shown in the following table:

Table 31 CDS Message Mediator Clients

| HDR 3.x Client System | Data Domains | Information Exchange Description | Information Element (Payload) |
| --- | --- | --- | --- |
| CHDR | * Allergy * Pharmacy | CHDR writes VIM-compliant allergy and pharmacy data directly to CDS. | Templates:   * PharmacyCreateOrUpdate40060.xsd * IntoleranceConditionCreateOrUpdate40060.xsd * createOrUpdateClinicalData   Message Exchange Standard: VIM  Payload Format: VIM |

#### Pathways SOAP Web Service Write Clients

Pathways provides a SOAP interface to be used by clients for logging sensitive patient data access. The information exchange details for this interchange are as specified in the following table:

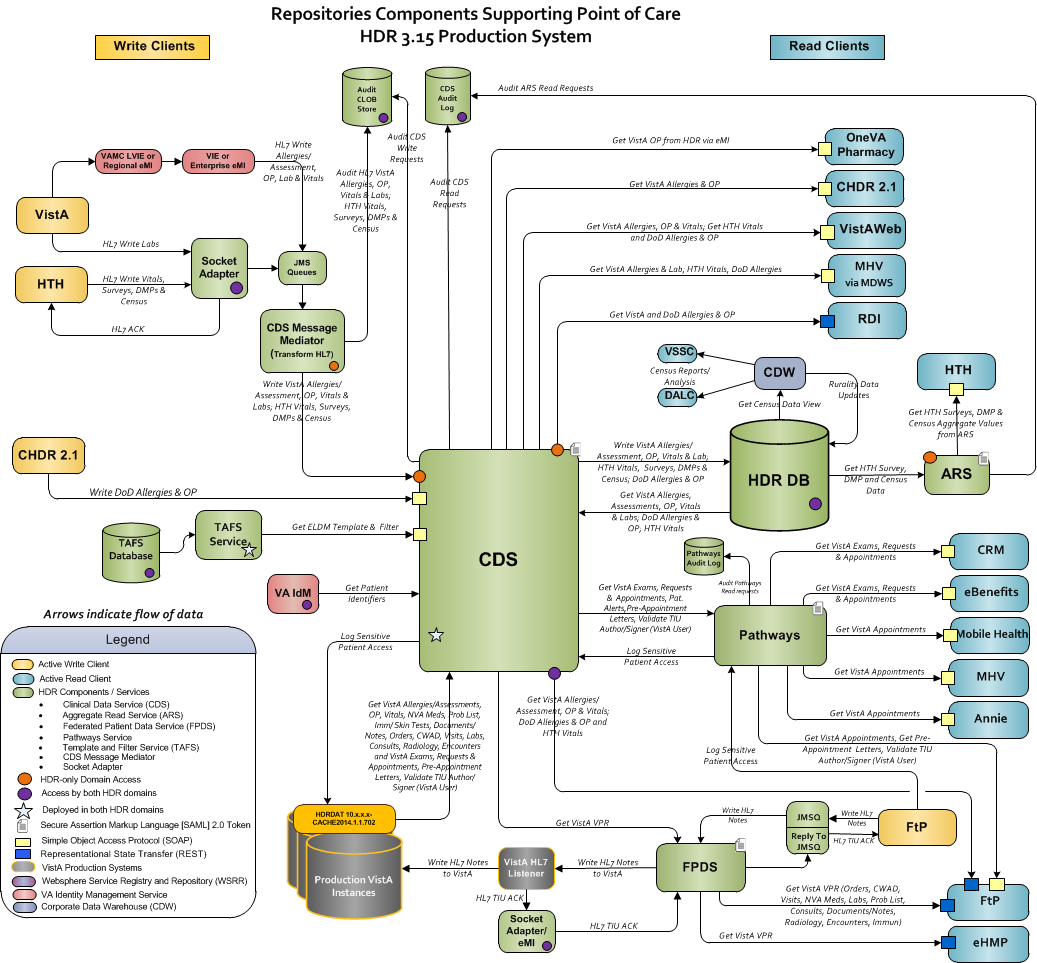
Table 32 Pathways SOAP Web Service Write Clients

| HDR 3.x Client System | Data Domains | Information Exchange Description | Information Element (Payload) |
| --- | --- | --- | --- |
| FtP | Sensitive Patient Log | FtP uses Pathways SOAP service directly to request the creation of a log entry in VistA DG Security Log.  The path parameters are used to construct a template as required by the CDS framework for creating data.  The framework will call a stored procedure in HDRDAT that will use the information to create an entry in the DG Security Log. Any errors will be returned using the standard CDS error response. | Template:   * SensitivePatientAccessCreate1   Payload Format: XML  VistA API Used: DG SENSITIVE RECORD ACCESS |

# External System Interface Design

The CDS process flow requires interaction with various enterprise and internal web services, data repositories, data access components and data store components as illustrated in the following figure:

Figure 18 CDS interactions with services/components/data repositories



## Interface Architecture

### CDS Modules Overview

The HDR ecosystem is comprised of CDS and supporting components, including: ARS, FPDS, Pathways, HDRDAT, and the HDR database itself. Other internal services include: TAFS, Auditing and Logging, and Error Handling services. The paradigm of Service Oriented Architecture is not only employed for interaction with other VA Enterprise Systems, it is extended for interaction within HDR’s various modules. The phrase CDS External Access is being used to explain the mechanism by which external VA Enterprise Systems interact with CDS, and the phrase CDS Internal Access to explain interaction amongst various modules within HDR.

#### CDS Internal Access

CDS accesses the data repositories (HDRDAT and HDR or HDR only) using the data sources configured in the WebLogic environment. CDS queries several instances of HDRDAT and queries/stores in HDR for the client request clinical data. CDS utilizes the Audit CLOB Store component to log the audit of the Read request along with the cursory information of the Read response in HDR. CDS is a Java EE application which will be deployed in a WebLogic cluster environment. The WebLogic clusters are configured with data sources of HDRDAT and HDR or HDR-Only. CDS uses Hibernate to retrieve/persist HDRDAT/HDR using the JDBC connections provided by the data sources. The Audit CLOB Store will use Hibernate to persist audits of the Read request/response using the JDBC connections provided by the HDR data sources.

The following table illustrates the data source details used by CDS in order to access the HDR/HDRDAT.

Table 33 CDS Data Sources

| Data Store | Interface | Input Messages | Output Messages |
| --- | --- | --- | --- |
| IdM / TAFS web services | eMI WSRR registry | CDS connects to the eMI WSRR service using JAXR, a standard API for interacting with web service repositories. Once the endpoint is resolved from the eMI WSRR service, a web service client is instantiated by the HDR client and injected with the endpoint Uniform Resource Identifier (URI) for the service. | The web service endpoint is used by CDS to evoke the required business API. |
| HDRDAT | JDBC data source pool for each of HDRDAT database instance is maintained in the HDR-only WebLogic cluster in which CDS is deployed. | CDS will use Spring to obtain the data source via JNDI lookup by passing the HDRDAT data source specific JNDI information as configured in WebLogic environment where CDS application is deployed. | The data source which provides the JDBC connection to HDRDAT database. Upon obtaining the JDBC connection to HDRDAT database, CDS will use Hibernate to query for clinical data by passing VistA site id/patient (local) identifier of the patient ICN and start and end dates specified in the clinical data request filter XML received as part of the client read request. The query response from HDRDAT will contain the patient (local) identifier-specific clinical data whose clinically relevant date is between the date ranges specified in the query. |
| HDR | JDBC data source pool for HDR database maintained in both the WebLogic clusters in which CDS is deployed. | CDS will use Spring to obtain the data source via JNDI lookup by passing the HDR data source specific JNDI information as configured in WebLogic environment where CDS application is deployed. | The data source which provides the JDBC connection to HDR database. Upon obtaining the JDBC connection to HDR database, CDS will use Hibernate to query for clinical data by passing VistA site id/patient (local) identifier of the patient ICN and start and end dates specified in the clinical data request filter XML received as part of the client read request. The query response from HDR will contain the patient (local) identifier specific clinical data whose clinically relevant date is between the date ranges specified in the query. |
| VistA HL7 Listener | HL7 listener for VistA maintained by VistA | TIU HL7 messages will be sent by FPDS via a socket connection to the HL7 listener that will validate the message, verify the provider and patient information and use the VistA TIU package to create an HL7 TIU Note in VistA. | Hl7 acknowledgements, responses and errors from VistA during processing are sent back to HDR embedded in a JMS message and deposited on a client designated response queue. |

#### eMI WSRR – Web Service Registry Service

HDR Service Endpoint Resolution is supported via eMI WebSphere Registry and Repository (WSRR) services. HDR components interact with the WSRR using the standard Java API for XML Registries (JAX-R API).

CDS gains access to the enterprise (IdM) and internal services (TAFS) by getting their connection information from eMI WSRR. According to enterprise requirements, the providers and consumers of the services must register/access the services from a registry service. eMI WSRR is a service registry that hosts the services and administers the run-time and deployment information of the services.

HDR incorporates the use of the eMI WSRR service for internally resolving service endpoint information at runtime for services consumed by HDR components. HDR components needing to perform an operation on a remote web service, connect to the eMI WSRR service using JAXR, a standard API for interacting with web service repositories. Once the endpoint is resolved from the eMI WSRR service, a web service client is instantiated by the HDR framework and injected with the endpoint Uniform Resource Identifier (URI) for the service into the application component. All clients are being asked to migrate to the eMI WSRR service as use of UDDI services hosted by the HDR will be terminated.

WSRR is just another directory service which contains end point information. It is not essential for CDS existing clients to look up CDS end point information from WSRR. The existing clients of CDS can connect to CDS directly without having to look up CDS info in WSRR. The VA enterprise encourages as part of an enterprise wide effort to move all projects to a Service Oriented Architecture that all application services are registered in a single directory.

CDS is a Java EE application which will be deployed in a WebLogic cluster environment. WebLogic cluster will be configured with data sources of HDRDAT and HDR or HDR only. CDS retrieves data from HDRDAT and retrieves/persists data in HDR using the JDBC connections provided by the data sources.

CDS accesses the Audit CLOB Store and HDR Oracle data stores using an Oracle JDBC RAC-enabled data source configured within the WebLogic server system. CDS accesses VistA data via the HDRDAT using JDBC data sources configured within the WebLogic server system.

##### SOAP requests with WSRR lookup

The SOAP remote procedure call (RPC) over HTTP client leverages the HTTP POST protocol and initiates SOAP structured message requests on the server using a well-known URL The web service then delegates the request to a stateless session bean for request fulfillment. The SOAP service is registered in WSRR.

When a read request arrives at the CDS interface component for processing, the application server receives the request over the well-known URL and identifies the operation name value; it passes it to the appropriate method on the web service interface. These requests are internally processed by CDS EJB component that is deployed on the application server as a stateless session bean. The EJB delegates the incoming requests to CDS framework.

##### SOAP requests without WSRR lookup

The SOAP remote procedure call (RPC) over HTTP client leverages the HTTP POST protocol and initiates SOAP structured message requests on the server using a well-known URL The web service then delegates the request to a stateless session bean for request fulfillment. This interface is exposed as a coarsely grained Web service to all clients. However, the Service Interface is not registered in WSRR. It provides clients functionality to read non-clinical data using the ELDM-formatted payloads for data retrieved from VistA systems. The component delegates client requests to framework objects which implement the logic to parse and fulfill the data request as specified by the ELDM filter payload.

##### REST web service

The REST web service client uses the HTTP protocol and leverages a collection of server side operations using the HTTP methods GET. The client can initiate an HTTP request to a server, which in turn processes the request and returns an appropriate response based on the HTTP method submitted. As with the SOAP client, the REST web service implementation logic on the server delegates to a stateless session bean for request fulfillment.

Service registration in eMI WSRR: The endpoint of the REST service is registered in eMI WSRR that is hosted on the WebLogic application server. The endpoints configuration is as follows and is in the classpath of WebLogic environment in which the services are deployed.

Figure 19 SOAP and REST server-side implementation, configuration of delegating objects

<jaxws:endpoint id="cdsService"

implementorClass="gov.va.med.cds.ejb.ClinicalDataServiceSynchronousSession"

implementor="#cdsStatelessSession"

address="/cds-service"

/>

<jaxrs:server id="createServiceProxy" address="/cds-service/createClinicalData">

<jaxrs:serviceBeans>

<bean class="gov.va.med.cds.webservice.RestClinicalDataServiceCreateProxy">

<property name="clinicalDataService" ref="cdsStatelessSession"/>

</bean>

</jaxrs:serviceBeans>

</jaxrs:server>

<jaxrs:server id="deleteServiceProxy" address="/cds-service/deleteClinicalData">

<jaxrs:serviceBeans>

<bean class="gov.va.med.cds.webservice.RestClinicalDataServiceDeleteProxy">

<property name="clinicalDataService" ref="cdsStatelessSession"/>

</bean>

</jaxrs:serviceBeans>

</jaxrs:server>

<jaxrs:server id="readServiceProxy" address="/cds-service/readClinicalData1">

<jaxrs:serviceBeans>

<bean class="gov.va.med.cds.webservice.RestClinicalDataServiceReadProxy">

<property name="clinicalDataService" ref="cdsStatelessSession"/>

</bean>

</jaxrs:serviceBeans>

</jaxrs:server>

<bean id="cdsStatelessSession"

class="org.springframework.ejb.access.LocalStatelessSessionProxyFactoryBean"

lazy-init="true">

<property name="lookupHomeOnStartup" value="false" />

<property name="jndiName">

<value>comp/env/ejb/local/gov/va/med/cds/ClinicalDataServiceSynchronousSession</value>

</property>

<property name="jndiEnvironment">

<props>

<prop key="java.naming.factory.initial">weblogic.jndi.WLInitialContextFactory</prop>

</props>

</property>

<property name="businessInterface">

<value>gov.va.med.cds.client.ClinicalDataServiceInterface</value>

</property>

</bean>

The RestClinicalDataServiceCreateProxy is passed the request by the application server, and delegates to the clinicalDataServiceSynchronous object, which is an implementation of the ClinicalDataServiceSynchronousSession EJB.

Figure 20 SOAP calls without WSRR:

<jaxws:endpoint id="pathways"

implementorClass="gov.va.med.cds.rpc.ejb.ClinicalDataServiceRpcSynchronousSession"

implementor="#pathwaysStatelessSession"

address="/pathways"

/>

<jaxrs:server id="readServiceProxy" address="/pathways/readData">

<jaxrs:serviceBeans>

<bean class="gov.va.med.pathways.webservice.RestPathwaysServiceReadDataProxy">

<property name="pathwaysService" ref="pathwaysStatelessSession"/>

</bean>

</jaxrs:serviceBeans>

</jaxrs:server>

<bean id="pathwaysStatelessSession" class="org.springframework.ejb.access.LocalStatelessSessionProxyFactoryBean" lazy-init="true">

<property name="lookupHomeOnStartup" value="false" />

<property name="jndiName">

<value>comp/env/ejb/local/gov/va/med/cds/ClinicalDataServiceRpcSynchronousSession</value>

</property>

<property name="jndiEnvironment">

<props>

<prop key="java.naming.factory.initial">weblogic.jndi.WLInitialContextFactory</prop>

</props>

</property>

<property name="businessInterface">

<value>gov.va.med.repositories.Pathways</value>

</property>

##### VistA HL7 Interface

The VistA HL7 interface uses VistA’s TIU HL7 Generic interface to write Notes to VistA. This requires HL7 components to be configured at each VistA site (Logical links, Sending Application, Receiving Application, Event Driver, and Event Subscriber). In addition, the HL7 message must comply with the data requirements outlined in the VistA Text Integration Utilities (TIU) Generic HL7 Interface Handbook.

The mechanics of the service is as follows. First, it makes a direct connection to the remote VistA site HL7 listener (logical link) and sends the HL7 message. The VistA listener does some initial checks and responds with an immediate HL7 acknowledgement (ACK),  Commit Accept (CA) or Commit Reject (CR). Those messages that receive a CA are queued up to be processed by the TIU message manager. The message manager performs some additional checks and replies with a second acknowledgement known as Application Accept (AA) or Application Reject (AR). Those messages receiving an AA will persist the Note into the VistA system.

### Mechanisms to interface CDS with external Clients

CDS interfaces with its clients through SOAP, REST, JMS and Socket-based services. A Write request submitted by the client application can be a VIM write XML instance, a JSON document, or an HL7 payload. The Read request can be a filter XML, a JSON document, an HL7 query message or a REST query string. The clinical data exchanged between CDS and its clients takes the form of XML documents representing VIM template instances or JSON documents. The format of these XML instances is governed by XSD schema definitions for VIM templates that are specific to the clinical domain. Refer to Data Exchanges in the following section.

* CDS exchanges VIM-based XML clinical/questionnaire results/expressions data with client applications as defined by the templates and filters.
* Pathways exchanges VIM-complaint non-clinical data with client applications as defined by the templates and filters.
* FPDS exchanges HL7, XML and JSON clinical and non-clinical data with client applications.
* For eHMP, CDS exchanges JSON formatted clinical data with eHMP.
* CDS exchanges XML HTH Census, DMP, ADL and Surveys data with HTH as defined by the templates and filters.
* HTH Census, DMP, ADL and Patient satisfaction survey data are generated by HTH vendors in HL7 format. They are sent to HDR via VA HTH system to be stored in the HDR data stores.
* Data exchange between CDS and client applications is confined to specific templates and filters. The filters listed in Table 33 are used for Read requests between CDS and client applications.

#### Data Exchanges

The following table includes client detail for domain entry point, applicable template(s) and associated filter IDs used in Read transactions between HDR services and client applications:

Table 34 Read Transactions – Domain Entry Point, Template and Filter

| **Client** | **Domain** | **Domain Entry Point** | | **Template(s)** | **Filter(s)** |
| --- | --- | --- | --- | --- | --- |
| **Aggregate Read Service (ARS)** | | | | | |
| HTH | ADLs | adlSurveyResponse | | ADLSurveyCreate40024 | HTREPORTFILTER.xsd |
| HTH | Census | CensusSurveyResponse | | CensusSurveyCreate40024 | HTREPORTFILTER.xsd |
| HTH | DMP | dmpSurveyResponse, dmpSurveyResponseWithScore | | DMPSurveyCreate40024 | HTREPORTFILTER.xsd |
| HTH | Patient Satisfaction Survey | psSurveyResponse | | PSSurveyCreate40025 | HTREPORTFILTER.xsd |
| **Clinical Data Service (CDS)** | | | | | |
| **Allergy / Intolerance Condition /Assessment** | | | | | |
|  | Allergy | intoleranceCondition | | AllergiesRead40010 | ALLERGY\_SINGLE\_PATIENT\_ALL\_DATA\_FILTER  CHDR\_ALLERGY\_SINGLE\_PATIENT\_FILTER |
| CHDR | Allergy | intoleranceCondition | | IntoleranceConditionRead40010 | CHDR\_ALLERGY\_SINGLE\_PATIENT\_FILTER |
| FtP | Allergy | intoleranceCondition | | RDIAllergiesPharmacyRead40013 | RDI\_ALLERGY\_RX\_SINGLE\_PATIENT\_FILTER |
| MHV | Allergy | intoleranceCondition | | MHVIntoleranceConditionRead40011.xsd | IC\_SINGLE\_PATIENT\_ALL\_DATA\_FILTER |
| RDI | Allergy/Rx/  Assessment | IntoleranceCondition OutpatientMedicationPromise AllergyAssessment | | RDIAllergiesPharmacyRead40013 | RDI\_ALLERGY\_RX\_SINGLE\_PATIENT\_FILTER |
| VW | Allergy |  | | VWAllergiesRead40010 | VW\_ALLERGY\_SINGLE\_PATIENT\_FILTER |
| VW | Allergy | intoleranceCondition | | VWIntoleranceConditionRead40010.xsd | VW\_IC\_SINGLE\_PATIENT\_FILTER |
| **Immunizations** | | | | | |
|  | Immunizations | immunizationEvents | | ImmunizationRead3 | IMMUNIZATION\_SINGLE\_PATIENT\_FILTER |
| **Labs** | | | | | |
| MHV | Labs | labTestPromise | | MHVLabRead40011 | LAB\_SINGLE\_PATIENT\_ALL\_DATA\_FILTER.XSD |
| **Non-VA Medications** | | | | | |
|  | NVA Meds | nonVaMedicationEvents | NonVAMedicationsRead2 | | NONVAMED\_SINGLE\_PATIENT\_FILTER |
| **Outpatient Pharmacy** | | | | | |
| CHDR | Outpatient Pharmacy | outpatientMedicationPromise | PharmacyRead40010 | | CHDR\_RX\_SINGLE\_PATIENT\_FILTER  RX\_SINGLE\_PATIENT\_ALL\_DATA\_FILTER  RX\_SINGLE\_PATIENT\_SINGLE\_RECORD\_FILTER |
| FtP | Outpatient Pharmacy | outpatientMedicationPromise | OutpatientPharmacyRead1 | | RX\_SINGLE\_PATIENT\_ALL\_DATA\_FILTER |
| OneVA | Outpatient Pharmacy | outpatientMedicationPromise | PharmacyRead40010 | | CHDR\_RX\_SINGLE\_PATIENT\_FILTER  RX\_SINGLE\_PATIENT\_ALL\_DATA\_FILTER  RX\_SINGLE\_PATIENT\_SINGLE\_RECORD\_FILTER |
| VW | Outpatient Pharmacy | outpatientMedicationPromise | VWPharmacyRead40010 | | RX\_SINGLE\_PATIENT\_ALL\_DATA\_FILTER  RX\_SINGLE\_PATIENT\_SINGLE\_RECORD\_FILTER |
| VW | Outpatient Pharmacy Detail | outpatientMedicationPromise | VWPharmacyDetailRead40010 | | RX\_SINGLE\_PATIENT\_ALL\_DATA\_FILTER  RX\_SINGLE\_PATIENT\_SINGLE\_RECORD\_FILTER |
| **Problem List** | | | | | |
|  | Problem List | healthConcerns | ProblemListRead2 | | PROBLEMLIST\_SINGLE\_PATIENT\_FILTER |
| **Skin Tests** | | | | | |
|  | Skin Test | skinTestProcedures | SkintestRead3 | | SKINTEST\_SINGLE\_PATIENT\_FILTER |
| **TIU Documents** | | | | | |
|  | TIU Detail | clinicalDocumentEvents | TiuDocumentDetailRead2 | | TIU\_SINGLE\_PATIENT\_DETAIL\_DATA\_FILTER |
|  | TIU List | clinicalDocumentEvents | TiuDocumentListRead2 | | TIU\_SINGLE\_PATIENT\_LIST\_DATA\_FILTER |
| **Vital Signs** | | | | | |
| VW | Vital Signs | vitalSignObservationEvent | VitalsignsRead40010 | | VITAL\_SINGLE\_PATIENT\_ALL\_DATA\_FILTER  VW\_VITAL\_SINGLE\_PATIENT\_FILTER |
| MHV | Vital Signs | vitalSignObservationEvent | MHVVitalsignsRead40010 | | MHV\_VITAL\_SINGLE\_PATIENT\_FILTER |
| **Federated Patient Data Service (FPDS Domain)** | | | | | |
| **Virtual Patient Record (VPR) via access to domains** | | | | | |
| FtP | Consults | GenericObservation | GenericObservationRead1  (FPDS Domain: CONSULT) | | GENERIC\_VISTA\_LIST\_DATA\_FILTER |
| FtP | Encounters | GenericObservation | GenericObservationRead1  (FPDS Domain: POV) | | GENERIC\_VISTA\_LIST\_DATA\_FILTER |
| FtP | Immunization | GenericObservation | GenericObservationRead1  (FPDS Domain: IMMUNIZA) | | GENERIC\_VISTA\_LIST\_DATA\_FILTER |
| FtP | Labs | GenericObservation | GenericObservationRead1  (FPDS Domain: LAB) | | GENERIC\_VISTA\_LIST\_DATA\_FILTER |
| FtP | Notes | GenericObservation | GenericObservationRead1  (FPDS Domain: DOCUMENT) | | GENERIC\_VISTA\_LIST\_DATA\_FILTER |
| FtP | NVA Meds | GenericObservation | GenericObservationRead1  FPDS Domain: MED; Type = OTC) | | GENERIC\_VISTA\_LIST\_DATA\_FILTER |
| FtP | Orders | GenericObservation | GenericObservationRead1  (FPDS Domain: ORDER) | | GENERIC\_VISTA\_LIST\_DATA\_FILTER |
| FtP | Patient Record Flags | GenericObservation | GenericObservationRead1  (FPDS Domain: FLAGS) | | GENERIC\_VISTA\_LIST\_DATA\_FILTER |
| FtP | Problems | GenericObservation | GenericObservationRead1  (FPDS Domain: PROBLEM) | | GENERIC\_VISTA\_LIST\_DATA\_FILTER |
| FtP | Progress Notes/Posting | GenericObservation | GenericObservationRead1  (FPDS Domain: DOCUMENT) | | GENERIC\_VISTA\_LIST\_DATA\_FILTER |
| FtP | Radiology | GenericObservation | GenericObservationRead1  (FPDS Domain: RAD) | | GENERIC\_VISTA\_LIST\_DATA\_FILTER |
| FtP | Visits | GenericObservation | GenericObservationRead1  (FPDS Domain: VISIT) | | GENERIC\_VISTA\_LIST\_DATA\_FILTER |
| **Pathways Service** | | | | | |
| **Appointments** | | | | | |
|  | Appointments | appointments | AppointmentsRead1 | | APPOINTMENTS\_SINGLE\_PATIENT\_FILTER |
| **Exams** | | | | | |
|  | Exams | Exam2507 | RequestsAndExamsRead2 | | REQUESTS\_AND\_EXAMS\_SINGLE\_PATIENT\_FILTER |
| **Exam Request** | | | | | |
|  | Exam Request | ExamRequest2507 | RequestsAndExamsRead2 | | REQUESTS\_AND\_EXAMS\_SINGLE\_PATIENT\_FILTER |
| **Patient Alerts** | | | | | |
| FtP | Patient Alert Detail | PatientAlert | PatientAlertDetailRead1 | | PATIENTALERT\_SINGLE\_PATIENT\_DETAIL\_FILTER |
| FtP | Patient Alert List | PatientAlert | PatientAlertListRead1 | | PATIENTALERT\_SINGLE\_PATIENT\_LIST\_FILTER |
| **VistA Pre-Appointment Letters** | | | | | |
| FtP | Pre-Appointment Letters | AppointmentLetter | AppointmentLetter1 | | APPOINTMENT\_LETTER\_SIMGLE\_PATIENT\_FILTER |
| **VistA Users** | | | | | |
| FtP | VistAUser | VistaUser | VistaUserRead1 | | FTP\_VISTA\_USER\_FILTER |

Table 35 Write Transactions – Templates

|  |  |  |  |
| --- | --- | --- | --- |
| **Client** | **Domain** | | **Write Template** |
| CHDR, VistA | | Pharmacy | PharmacyCreateOrUpdate40060.xsd |
| CHDR, VistA | | Allergy | IntoleranceConditionCreateOrUpdate40060.xsd |
| VistA | | Allergy Assessment | Allergy AssessmentCreateOrUpdate40060.xsd |
| VistA, HTH | | Vital Signs | VitalsignsCreateOrUpdate40060.xsd |
| VistA | | Lab | LabCreateOrUpdate40060.xsd |
| HTH | | Census | CensusSurveyCreate40024.xsd |
| HTH | | DMP | DMPSurveyCreate40024.xsd |
| HTH | | ADL | ADLSurveyCreate40024.xsd |
| HTH | | PSS | PSSurveyCreate40023.xsd (PS1.0) |
| HTH | | PSS | PSSurveyCreate40025.xsd (PS2.0) |
| FtP | | TIU | N/A (HL7) |
| FtP (via Pathways) | | Log Sensitive Patient Access | SensitivePatientAccessCreate1.xsd |

### Client Interface Detailed Design

The HDR system provides a synchronous Java Application Programming Interfaces (API) for XML-based Web Services (JAX-WS) and Java API for RESTful Web Services (JAX-RS) web services using the Apache CXF open source services framework; an asynchronous application programming interface using the Java Messaging Service (JMS) and a TCP/IP socket interface using XSockets.

* HDR system clients can connect to the Java API services using either a Simple Object Access Protocol (SOAP) or a Representational State Transfer (REST) Web service interface. CDS, ARS and Pathways clients can connect to the service using SOAP web service interface and FPDS clients can connect to the service using a REST web service interface.
* VIM clinical data Read clients connect synchronously to CDS and receive a VIM response payload on the same connection. CDS, Pathways and ARS Read clients use this feature of the HDR system for obtaining VIM data from HDR DB and/or VistA.
* FPDS Read clients connect synchronously to VistA for VPR data.
* Clinical data Write clients are able to persist clinical data to HDR for the following VistA clinical domains: Allergies, Vitals, Outpatient Pharmacy (OP), and Laboratory. The clients use either the JMS interface or the Socket interface for data exchange. CDS Message Mediator and FDPS expose a JMS interface while the CDS Socket adapter exposes a socket interface to clients.
* HL7 write clients connect to either the CDS Message Mediator or CDS Socket Adapter for persisting data to HDR database. The Socket Adapter is a client of the CDS Message Mediator and places all HL7 messages it receives on the CDS Message Mediator JMS queues. The HL7 message is transformed to VIM by the CDS Message Mediator prior to persistence to the HDR database.
* HL7 write clients connect to FPDS JMS interface for persisting data to VistA through the VistA HL7 Listener.

#### CDS Data Service Interface

The CDS component features the CDS Data Service Interface component that is:

* Exposed to service requests for clinical data from VistA systems/HDR. This interface is exposed as a coarsely grained Web service to all clients. The Service Interface is registered in eMI WSRR.
* Exposed to service requests for non-clinical data from VistA systems. This interface is exposed as a coarsely grained Web service to all clients. However, the Service Interface is not registered in eMI WSRR. It provides clients functionality to read non-clinical data using the VIM-formatted payloads for data retrieved from VistA systems. The component delegates client requests to framework objects which implement the logic to parse and fulfill the data request as specified by the VIM filter payload.
* Exposed as a coarsely grained RESTful Web Service and is accessed as a URL.

#### VIM Reads

VIM clinical data Read clients submit Read requests for clinical data results. The Read request is maintained for auditing purposes. All applicable data sources are queried and results are returned to the client in the appropriate payload structure.

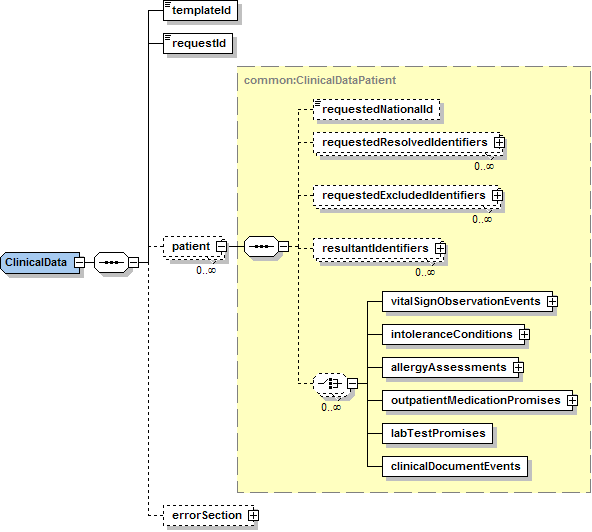
##### CDS Read Request

A Read request is used by the VIM read clinical data client to obtain patient-centric clinical data. The input parameters for the Read request are as follows:

* The template ID specifies the type of data payload to be returned to the client application. Template IDs are defined for each supported clinical domain. These include the OP, Allergies, Vitals, Text Integration Utilities (TIU), and Laboratory (Chemistry/Hematology) clinical domains.
* The filter request is an XML document used by CDS and Pathways to identify the specific data retrieval conditions. For Read requests, the filter request includes the patient identification information, as well as an optional date range.
* The filter ID designates the schema used by CDS and Pathways to define and validate filter requests.
* The request ID is a unique identifier provided by the requesting site that is used to identify the read request within the system.
* The string value returned by this request is an XML document that is an instance of the VIM template ID input parameter.

The Read response template contains the clinical data returned from the data sources and the errors generated while processing the request. Below is a general example of a Read response, a basic pattern that all specific templates follow.

Figure 21 Read Response Template



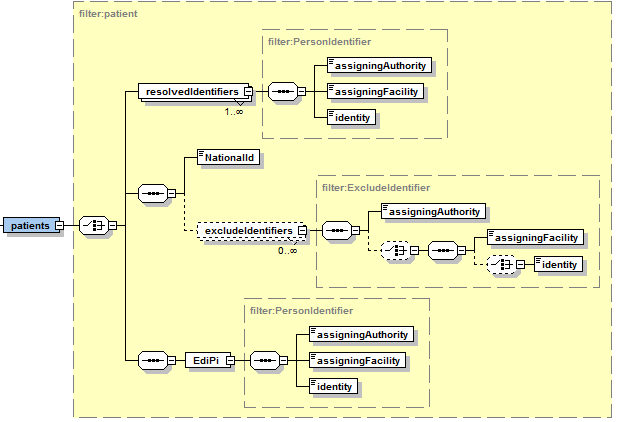
##### Filters

The filter determines which records are to be returned to the client. The filter is presented as an XML document. The first filter section defines the patient identification information that is used to qualify the data to be returned in the response.

* Patient identifiers may be specified by a single national identifier. CDS 3.x supports both the Integration Control Number (ICN) or the Electronic Data Interchange Person Identifier (EDIPI) as the national identifier. CDS 3.x or Pathways will perform a lookup to resolve the national identifier to a list of all local identifiers. Optionally, a list of exclude (local) identifiers can be specified to further restrict the information returned in the response.

The following figure shows the patient identification section in a filter:

Figure 22 Patient Identification in a Filter



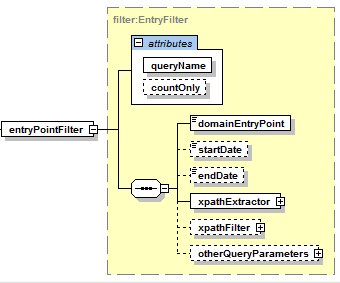
The second section specifies one or more entry point filters, which consist of the following:

* **Domain Entry Point**: A mandatory parameter that determines the type of data to be returned in the result document, illustrated in the following table. If the domainEntryPoint in the request filter is null/blank, CDS 3.x cannot determine the proper schema to include in the response document and a generic response will be sent to the client. Refer to Appendix B.1 for CDS Supported Domain Entry Points.
* **Start Date**: An optional parameter that represents the beginning point in time to be used to constrain the results.
* **End Date**: An optional parameter that represents the ending point in time to be used to constrain the results.
* **Record Identifier**: An optional list of parameters that is used to return a specific set of one or more records in the result.
* **XPath Query**: An optional parameter that consists of an XPath expression specified in the filter that can be applied to the resultant document to constrain the result.
* **Other Query parameters**: An optional list of one or more parameters that can be used to constrain the returned results. These optional query parameters can be used for ad-hoc query capability.
* **Query Timeout (Seconds):** The optional queryTimeoutSeconds parameter in the Read request filter, which specifies how many seconds CDS allows to pass before the Read request times out.

1. In the event that a reasonable date range is not specified in the filter, the responsiveness of the request could decrease and the size of the payload may be extremely large.

The following figure shows the entry point section of a filter:

Figure 23 Entry Point Section of a Filter



##### Error Responses

When processing results in errors that interrupt and abort Write or Read requests, such as validation, system errors, missing parameters, etc., the interface to CDS and FPDS will return a response that includes an error section. The error section contains detailed error explanations of the issues encountered. A sample eHMP XML response with an error section is illustrated in the following figure.

Figure 24 eHMP XML Response with Error Section

<?xml version="1.0" encoding="UTF-8"?>

<clinicaldata:ClinicalData xmlns:clinicaldata="Clinicaldata">

<templateId>GenericObservationRead1</templateId>

<requestId>65756756</requestId> -<patients> -<patient>

<requestedResolvedIdentifiers/> -<resultantIdentifiers> -<resultantIdentifier>

<identity>xxxxxxxxxxx</identity>

<assigningFacility>xxx</assigningFacility>

<assigningAuthority>USVHA</assigningAuthority>

</resultantIdentifier>

</resultantIdentifiers>

<genericObservations/>

</patient>

</patients> -<errorSection> -<fatalErrors> -<fatalError>

<errorId>65756756</errorId>

<exception>gov.va.med.repositories.fpds.validator.DefaultReadRequestValidator$InvalidParameterValidationError</exception>

<exceptionMessage>FPDS\_ERROR</exceptionMessage>

<errorCode>FPDS\_ERROR</errorCode>

<displayMessage>Request parameter clientName was not provided or was empty.</displayMessage>

</fatalError>

</fatalErrors>

<errors> </errors>

<warnings> </warnings>

</errorSection>

</clinicaldata:ClinicalData>

#### Federated Patient Data Service (FPDS) REST Interface

FPDS exposes VPR Subscribe-Get and VPR Patient-Get web service interfaces that provide data in either XML or JSON from VistA.

##### FPDS VPR Subscribe-Get REST Web Service interface

This interface provides the ability to subscribe patients to VPR VistA server cache and obtain updates to VPR patient data from VistA sites. Clients can use the cancel to unsubscribe patients from the VPR VistA server cache. Figure 25 shows the details of the interface that can be accessed from the resource base of [host:port]/repositories.med.DNS /fpds/vpr. Each operation has specific mandatory and optional query parameters as shown in the tables below.

This interface provides the ability to get patient specific VPR domain data, as illustrated in Figure 25.

Figure 25 FPDS VPR Subscribe-Get REST Web Service Interface

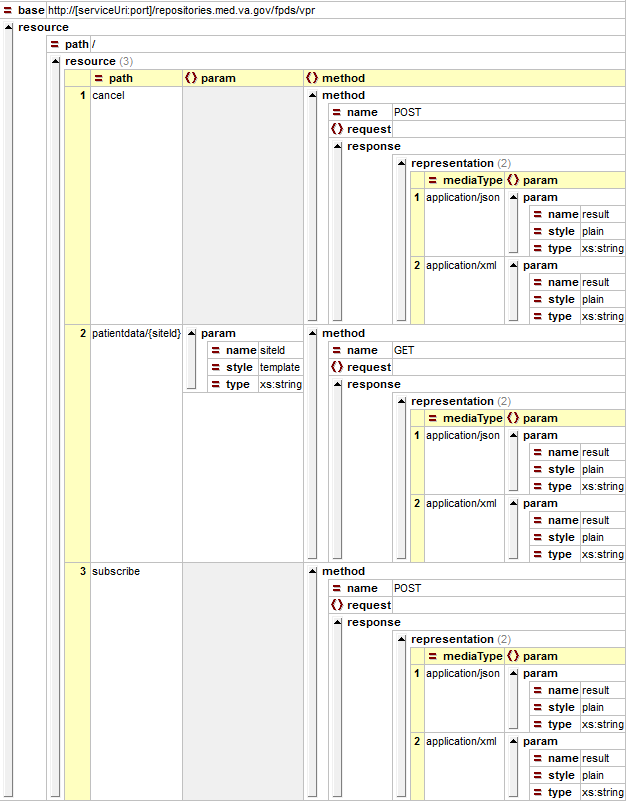


Table 36 Mandatory and Optional Query Parameters - Subscribe

| **Path Parameter(s)** | **Required Query Parameter(s)** | **Optional Query Parameter(s)** |
| --- | --- | --- |
| Base: [host:port]/repositories.med.DNS /fpds/vpr/subscribe | | |
|  | * clientName: name of the client requesting the data. * clientRequestInitiationTime: the timestamp that denotes the time at which the client had issued the request. * requestId: A unique alpha-numeric string assigned to the request for audit purposes * server: name of the VistA VPR server cache on VistA to which to subscribe * nationalId or resolvedIdentifier: The patient’s ICN or resolvelIdentior used when nationalId is not provided. resolvedIdentifier is used to query for data from a particular facility. This can be a comma separated list to get data from multiple specific facilities. resolvedIdentifier format: dfn-facilityid-assigningAuthority . | \_type: Type of the response format. Defaults to json. |
| Subscribe Examples: | | |
| 1. JSON with national Id:    1. [host:port]/repositoties.med.DNS /fpds/vpr/subscribe?clientName=eHMP&clientRequestInitiationTime=2015-04-09T12%3A16%3A08-06%3A00&requestId=1234-unique-id&server=HMPServer&nationalId=1008523096V381537 2. XML with national Id:    1. [host:port]/repositoties.med.DNS /fpds/vpr/subscribe?clientName=eHMP&clientRequestInitiationTime=2015-04-09T12%3A16%3A08-06%3A00&requestId=1235-unique-id&server=HMPServer&nationalId=1008523096V381537&\_type=xml 3. JSON with single resolvedIdentifier:    1. [host:port]/repositoties.med.DNS /fpds/vpr/subscribe?clientName=eHMP&clientRequestInitiationTime=2015-04-09T12%3A16%3A08-06%3A00&requestId=1236-unique-id&server=HMPServer&resolvedIdentifier=12345-666-USVHA&\_type=json 4. XML with multiple resolvedIdentifier: 5. [host:port]/repositoties.med.DNS /fpds/vpr/subscribe?clientName=eHMP&clientRequestInitiationTime=2015-04-09T12%3A16%3A08-06%3A00&requestId=1237-unique-id&server=HMPServer&resolvedIdentifier=12345-666-USVHA,34789-667-USVHA&\_type=xml | | |

Table 37 Mandatory and Optional Query Parameters – Patient Data

| **Path Parameter(s)** | **Required Query Parameter(s)** | **Optional Query Parameter(s)** |
| --- | --- | --- |
| Base: [host:port]/repositories.med.DNS /fpds/vpr/patientdata | | |
| {siteId}: VistA site from which VPR data updates will be retrieved | * clientName: name of the client requesting the data. * clientRequestInitiationTime: the timestamp that denotes the time at which the client had issued the request. * requestId: A unique alpha-numeric string assigned to the request for audit purposes * server: name of the VistA VPR server cache on VistA to which to subscribe * extractSchema: VistA VPR extract schema information * max: The maximum number of records to return. Values greater than 500 are reset to 500 * lastUpdate: Token used to identify the last row that was returned the VistA VPR server cache Format: vista date string-row number | \_type: Type of the read response format. Defaults to json. |
| Patient Data Examples: | | |
| 1. JSON:    1. [host:port]/repositoties.med.DNS /fpds/vpr/patientdata/{666}?clientName=eHMP&clientRequestInitiationTime=2015-04-09T12%3A16%3A08-06%3A00&requestId=1238-unique-id&server=HMPServer&extractSchema=3.001&max=489&lastUpdate=3150217-282 2. XML with embedded VPR JSON:    1. [host:port]/repositoties.med.DNS /fpds/vpr/patientdata/{666}?clientName=eHMP&clientRequestInitiationTime=2015-04-09T12%3A16%3A08-06%3A00&requestId=1239-unique-id&server=HMPServer&extractSchema=3.001&max=489&lastUpdate=3150217-282&\_type=xml | | |

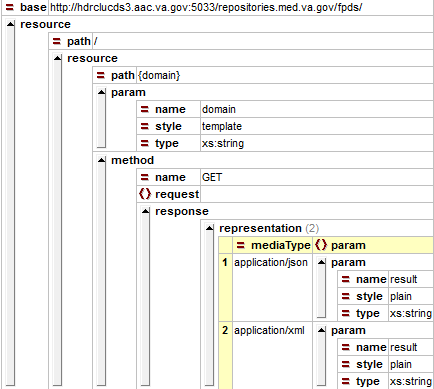
Table 38 Mandatory and Optional Query Parameters – Cancel

| **Path Parameter(s)** | **Required Query Parameter(s)** | **Optional Query Parameter(s)** |
| --- | --- | --- |
| Base: [host:port]/repositories.med.DNS /fpds/vpr/cancel | | |
|  | * clientName: name of the client requesting the data. * clientRequestInitiationTime: the timestamp that denotes the time at which the client had issued the request. * requestId: A unique alpha-numeric string assigned to the request for audit purposes * server: name of the VistA VPR server cache on VistA to which to subscribe * nationalId or resolvedIdentifier: The patient’s ICN or resolvelIdentior used when nationalId is not provided. resolvedIdentifier is used to query for data from a particular facility. This can be a comma separated list to get data from multiple specific facilities. resolvedIdentifier format: dfn-facilityid-assigningAuthority . | \_type. Type of the response format. Defaults to json. |
| Cancel Examples: | | |
| 1. JSON with national Id:    1. [host:port]/repositoties.med.DNS /fpds/vpr/cancel?clientName=eHMP&clientRequestInitiationTime=2015-04-09T12%3A16%3A08-06%3A00&requestId=2234-unique-id&server=HMPServer&nationalId=1008523096V381537 2. XML with national Id:    1. [host:port]/repositoties.med.DNS /fpds/vpr/cancel?clientName=eHMP&clientRequestInitiationTime=2015-04-09T12%3A16%3A08-06%3A00&requestId=2235-unique-id&server=HMPServer&nationalId=1008523096V381537&\_type=xml 3. JSON with single resolvedIdentifier:    1. [host:port]/repositoties.med.DNS /fpds/vpr/cancel?clientName=eHMP&clientRequestInitiationTime=2015-04-09T12%3A16%3A08-06%3A00&requestId=2236-unique-id&server=HMPServer&resolvedIdentifier=12345-666-USVHA&\_type=json 4. XML with multiple resolvedIdentifier:    1. [host:port]/repositoties.med.DNS /fpds/vpr/cancel?clientName=eHMP&clientRequestInitiationTime=2015-04-09T12%3A16%3A08-06%3A00&requestId=2237-unique-id&server=HMPServer&resolvedIdentifier=12345-666-USVHA,34789-667-USVHA&\_type=xml | | |

##### FPDS VPR Patient-Get REST Web Service interface

Figure 26 illustrates the details of the interface that can be accessed from the resource base of [host:port]/repositories.med.DNS /fpds/.

Figure 26 FPDS VPR Patient-Get REST Web Service Interface



The specific mandatory and optional query parameters for the FPDS Patient-Get interface are shown in the tables below.

Table 39 Mandatory and Optional Query Parameters – FPDS Patient-Get

| **Path Parameter(s)** | **Required Query Parameter(s)** | **Optional Query Parameter(s)** |
| --- | --- | --- |
| Base: [host:port]/repositories.med.DNS /fpds/ | | |
| {domain}: The clinical or non-clinical domain name. See VistA VPR Technical manual for supported domains. | * clientName: name of the client requesting the data. * clientRequestInitiationTime: the timestamp that denotes the time at which the client had issued the request. * requestId: A unique alpha-numeric string assigned to the request for audit purposes * nationalId or resolvedIdentifier: The patient’s ICN or resolvelIdentior used when nationalId is not provided. resolvedIdentifier is used to query for data from a particular facility. This can be a comma separated list to get data from multiple specific facilities. resolvedIdentifier format: dfn-facilityid-assigningAuthority . | * \_type. Type of the read response format. Defaults to json. * max: The maximum number of records to return. * text: Boolean to include document text in the response - true/false. * startDate: The start date of search in yyyy-MM-dd format * endDate: The end date of search in yyyy-MM-dd format. * id: The ID of the single record to return * excludeIdentifier : This is optional. If provided, requires the nationalId. It is used to exclude data from a facility. Format: dfn-facilityid-assigningAuthority. |
| Patient-Get Examples: | | |
| 1. National id URL with optional exclude identifier(s):   http://<host>:<port>/repositories.med.DNS /fpds/<domain>?clientName=HMP&clientRequestInitiationTime=<timestamp>&requestId=<requestId>&nationalId=<ICN>&excludeIdentifier=<dfn>-<assigningFacility>-<assigningassigningAuthority>&excludeIdentifier=<dfn>-<assigningFacility>-<assigningassigningAuthority>&max=<max>&startDate=<yyyy-MM-dd>&endDate=<yyyy-MM-dd>&text=<text>&id=<id>   1. Resolved identifier(s) URL:   http://<host>:<port>/repositories.med.DNS /fpds/<domain>?clientName=HMP&clientRequestInitiationTime=<timestamp>&requestId=<requestId>&resolvedIdentifier=<dfn>-<assigningFacility>-<assigningassigningAuthority>&resolvedIdentifier=<dfn>-<assigningFacility>-<assigningassigningAuthority>&max=<max>&startDate=<yyyy-MM-dd>&endDate=<yyyy-MM-dd>&text=<text>&id=<id>   1. JSON request with national id and exclude identifier(s):   http://<host>:<port>/repositories.med.DNS /fpds/PROBLEM?nationalId=1111&clientName=HMP&excludeIdentifier=2-666-USDOD&excludeIdentifier=333-100-USVHA&startDate=1990-01-01&endDate=2000-01-01&clientRequestInitiationTime=2014-05-30T09:30:10.5&max=100&text=true&id=140&requestId=111111   1. JSON request with resolved id(s):   http://<host>:<port>/repositories.med.DNS /fpds/PROBLEM?clientName=HMP&resolvedIdentifier=2-666-USVHA&resolvedIdentifier=333-100-USVHA&startDate=1990-01-01&endDate=2000-01-01&clientRequestInitiationTime=2014-05-30T09:30:10.5&max=100&text=true&id=222&requestId=111112   1. Read request URL format that needs the read response in VIM XML format:   http://<host>:<port>/repositories.med.DNS /fpds/PROBLEM?clientName=HMP&resolvedIdentifier=2-666-USVHA&resolvedIdentifier=333-100-USVHA&startDate=1990-01-01&endDate=2000-01-01&clientRequestInitiationTime=2014-05-30T09:30:10.5&max=100&text=true&id=222&requestId=111113&\_type=xml | | |

#### ARS Reads

ARS supports aggregate read services on data stored in the HDR DB for Census, Activities of Daily Living (ADL VR-12) surveys, Patient Satisfaction Surveys and DMPs. The ARS is a SOAP web service exposed over HTTPS that provides an aggregated (report) view of HTH data stored in the HDR DB. The support for Census reads through ARS consists of:

* Generating reports on HTH Census data stored in the HDR database.
* Providing calculations on the stored data and returning counts and values in the results report.
* Materialized view to be developed in the database layer.
* Report filter and report response template that are part of the production HDR system will be used for all Census aggregate reads

##### ARS Read Request

A Read request is used by the ARS read aggregate data client to obtain aggregated report data. The input parameters for the Read request are as follows:

* The filter request is an XML document used by ARS to identify the specific data retrieval conditions as well as aggregation options.
* The report ID is a unique report identifier used to identify what report is being requested.
* The client ID designates the client making the request.
* The request ID is a unique identifier provided by the requesting site that is used to identify the read request within the system.

##### Filters

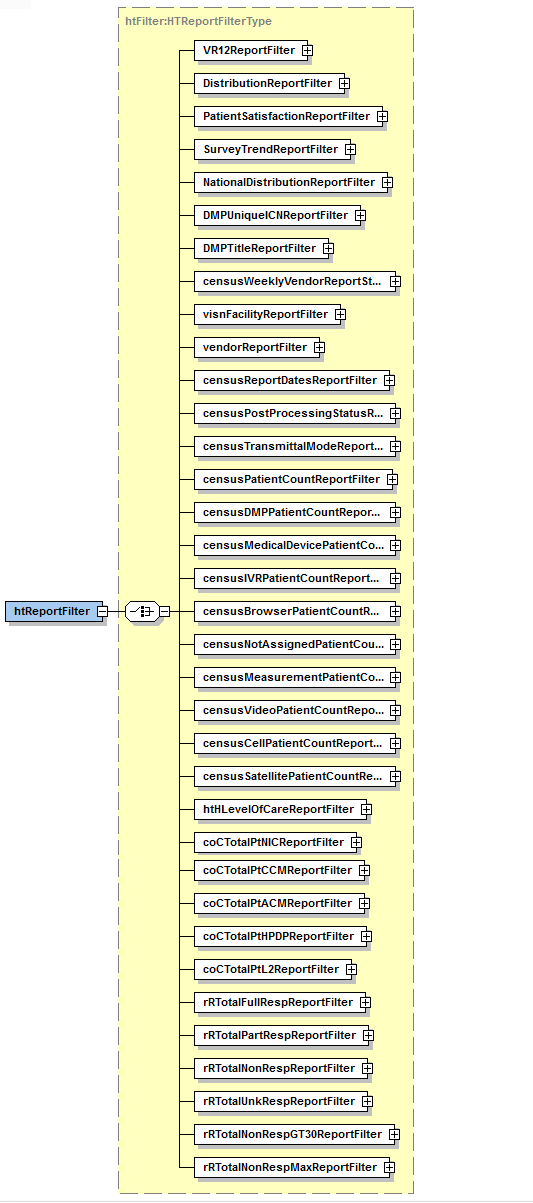
The filter determines which records are to be returned to the client. The filter is presented as an XML document. All filters must contain a reportID defined by the enumeration of values in Table 40. Filters may contain startDate and endDate elements that define the date range used to constrain a report. Filters may contain a location and vendor element that defines how the data is aggregated/grouped.

Table 40 ARS Report Identifiers

| **Report Identifiers** | |
| --- | --- |
| VR-12\_STATISTICS | HTH\_TRANSMITTALMODE\_REPORT |
| DISTRIBUTION\_OF\_SURVEYS | CENSUS\_TOTAL\_REPORT |
| PATIENT\_SATISFACTION\_STATISTICS | CENSUS\_DMPTOTAL\_REPORT |
| SURVEY\_TRENDS | CENSUS\_MEDDEVTOTAL\_REPORT |
| NATIONAL\_DISTRIBUTION\_OF\_SURVEYS | CENSUS\_IVRTOTAL\_REPORT |
| DMP\_UNIQUE\_ICN\_REPORT | CENSUS\_BROWSERTOTAL\_REPORT |
| DMP\_TITLE\_REPORT | CENSUS\_NOTASSIGNEDTOTAL\_REPORT |
| VISN\_FACILITY\_REPORT | CENSUS\_MEASUREMENTTOTAL\_REPORT |
| VENDOR\_REPORT | CENSUS\_VIDEOTOTAL\_REPORT |
| CENSUS\_WEEKLY\_VENDOR\_STATUS\_REPORT | CENSUS\_CELLMODEMTOTAL\_REPORT |
| CENSUS\_START\_END\_DATES\_REPORT | CENSUS\_SATELLITETOTAL\_REPORT |
| CENSUS\_POST\_PROCESSING\_STATUS\_REPORT | HTH\_LEVELOFCARE\_REPORT |
| COC\_TOTALPTNIC\_REPORT | COC\_TOTALPTCCM\_REPORT |
| COC\_TOTALPTACM\_REPORT | COC\_TOTALPTHPDP\_REPORT |
| COC\_TOTALPTL2\_REPORT | RR\_TOTALFULLRESP\_REPORT |
| RR\_TOTALPARTRESP\_REPORT | RR\_TOTALNONRESP\_REPORT |
| RR\_TOTALUNKRESP\_REPORT | RR\_TOTALNONRESPGT30\_REPORT |
| RR\_TOTALNONRESPMAX\_REPORT | CENSUS\_UNKNOWNFACILITY\_ADC\_REPORT |
| CENSUS\_URBANFACILITY\_ADC\_REPORT | CENSUS\_HIGHLYRURALFACILITY\_ADC\_REPORT |
| CENSUS\_RURALFACILITY\_ADC\_REPORT | CENSUS\_UNKNOWNVISN\_ADC\_REPORT |
| CENSUS\_UNKNOWNVISN\_ADC\_REPORT | CENSUS\_URBANVISN\_ADC\_REPORT |
| CENSUS\_HIGHLYRURALVISN\_ADC\_REPORT | CENSUS\_RURALVISN\_ADC\_REPORT |
| CENSUS\_TOTAL\_RURALVISN\_ADC\_REPORT | CENSUS\_NIC\_ADC\_REPORT |
| HTH\_FYQTR\_REPORT | HTH\_CUMULATIVEMONTHYEAR\_REPORT |

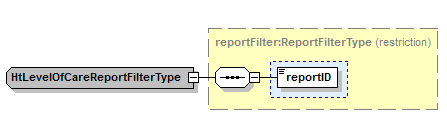
The following figure shows the complete ARS Report filter for current and new reports:

Figure 27 ARS Report Filter



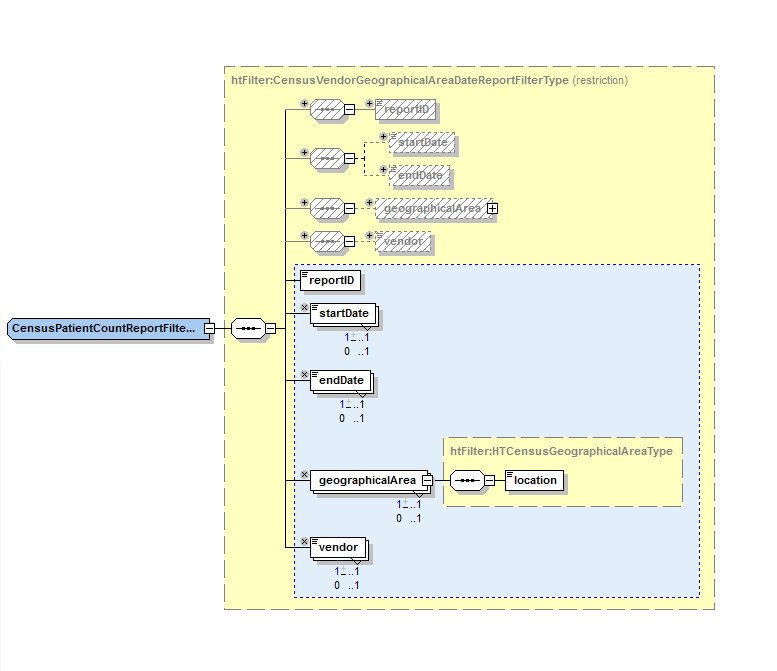
The following figure shows the complete HTH Level of Care Report filter which is unique to this specific report:

Figure 28 HTH Level of Care Report Filter



The following figure shows the complete Category of Care (COC) - Total Patient Count of Non-institutionalized Care (NIC) Report filter used for all other new census reports, except the one shown above:

Figure 29 Category of Care (COC) - Total Patient Count of Non-institutionalized Care (NIC) Report Filter



* **Report ID**: A required parameter that designates what report is being generated as listed in the ARS Report Identifiers table shown above.
* **Start Date**: A required parameter for most Census reports that represents the beginning point in time to be used to constrain the results. Date elements must conform to the “YYYY-MM-DD” format.
* **End Date**: A required parameter for most Census reports that represents the ending point in time to be used to constrain the results. Date elements must conform to the “YYYY-MM-DD” format.
* **Location**: A required parameter for most Census reports that defines how the data will be aggregated/grouped in the results. The location can be one of “National”, “All VISNS”, “VISN number” (example: “VISN 99”), “All VistA Facilities” or “VISN-VISN Facility Identifier” (example: “VISN 99-999AA”).
* **Vendor**: A required parameter for most Census reports that defines how the data will be aggregated/grouped in the results. The vendor component can be “All Vendors” or a specific vendor number (example: “200T1”).

#### Read Response

The Read response template contains the aggregate report data returned or the errors generated while processing the request. Below is a general example of an ARS Read response, a basic pattern that all specific templates follow.

Figure 30 Read Response Template

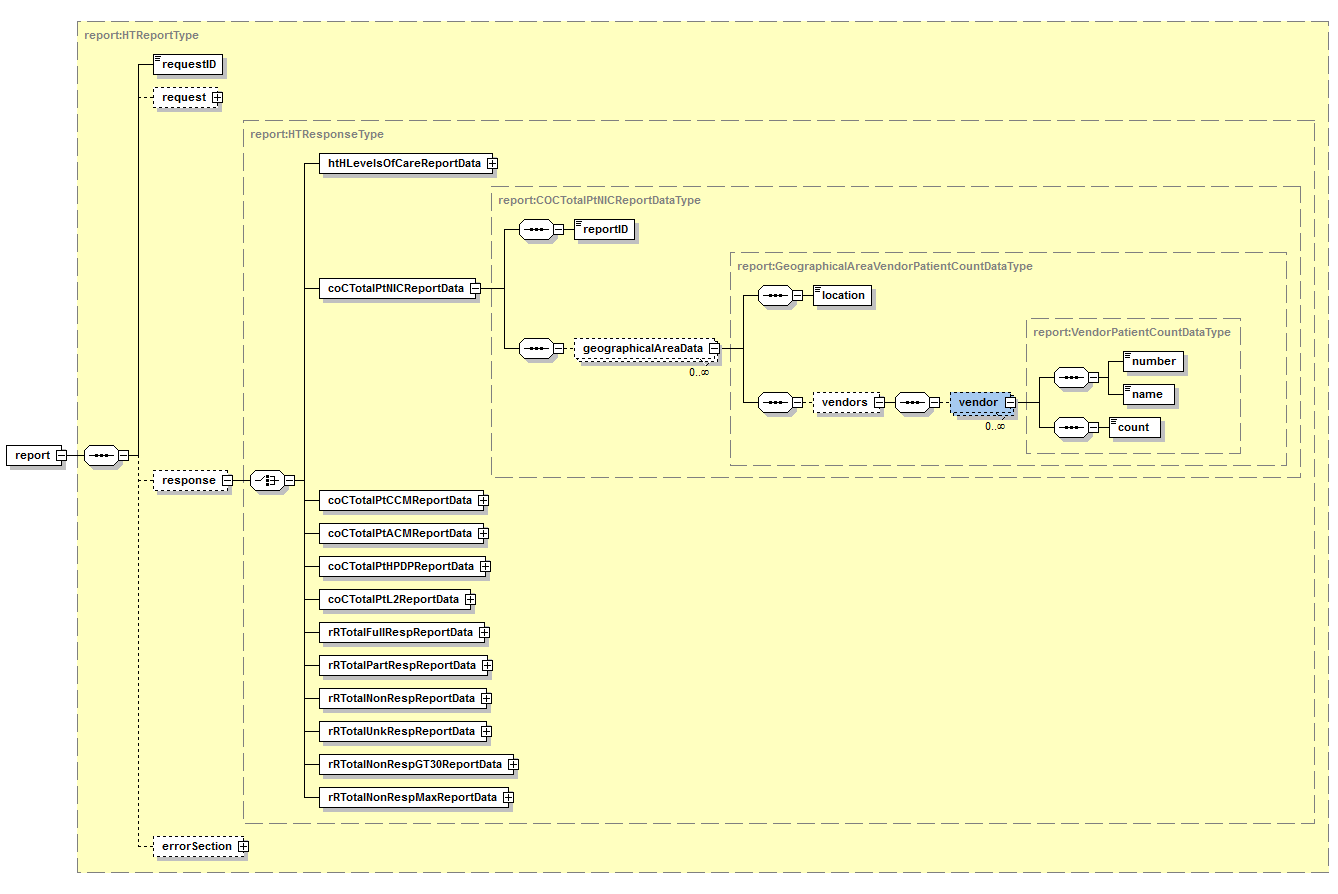


Figure 31 Example ARS XML Response

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<report:report xmlns:htFilter="HTReportFilter" xmlns:report="Report" xmlns:reportFilter="ReportFilter">

<requestID>MH8\_BROWSERTOTAL\_ALLVISTAFACIL\_SNGLVEND</requestID>

<request>

<censusBrowserPatientCountReportFilter>

<reportID>CENSUS\_BROWSERTOTAL\_REPORT</reportID>

<startDate>2015-01-18</startDate>

<endDate>2015-01-24</endDate>

<geographicalArea>

<location>All VistA Facilities</location>

</geographicalArea>

<vendor>200T7</vendor>

</censusBrowserPatientCountReportFilter>

</request>

<response>

<censusBrowserPatientCountReportData>

<reportID>CENSUS\_BROWSERTOTAL\_REPORT</reportID>

<geographicalAreaData>

<location>VISN 1-402</location>

<vendors>

<vendor>

<number>200T7</number>

<name>Telehealth Cardiocom</name>

<count>1</count>

</vendor>

</vendors>

</geographicalAreaData>

<geographicalAreaData>

<location>VISN 1-523A5</location>

<vendors>

<vendor>

<number>200T7</number>

<name>Telehealth Cardiocom</name>

<count>1</count>

</vendor>

</vendors>

</geographicalAreaData>

<geographicalAreaData>

<location>VISN 1-608</location>

<vendors>

<vendor>

<number>200T7</number>

<name>Telehealth Cardiocom</name>

<count>1</count>

</vendor>

</vendors>

</geographicalAreaData>

<geographicalAreaData>

<location>VISN 5-512</location>

<vendors>

<vendor>

<number>200T7</number>

<name>Telehealth Cardiocom</name>

<count>1</count>

</vendor>

</vendors>

</geographicalAreaData>

</censusBrowserPatientCountReportData>

</response>

<errorSection/>

</report:report>

##### Error Responses

When processing results in errors that interrupt and abort Write or Read requests, such as validation, system errors, missing parameters, etc., the interface to CDS and ARS will return a response that includes an error section. The error section contains detailed error explanations of the issues encountered. A sample ARS XML response with an error section is illustrated in the following figure.

Figure 32 ARS XML Response with Error Section

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<report:report xmlns:htFilter="HTReportFilter" xmlns:report="Report" xmlns:reportFilter="ReportFilter">

<requestID>TT\_TOTAL\_PT</requestID>

<request>

<censusPatientCountReportFilter>

<reportID>CENSUS\_TOTAL\_REPORT</reportID>

<startDate>2005-10-08</startDate>

<endDate>2005-10-15</endDate>

<geographicalArea>

<location>All VistA Facilities</location>

</geographicalArea>

<vendor>All Vendors</vendor>

</censusPatientCountReportFilter>

</request>

<errorSection>

<fatalErrors>

<errorId>TT\_TOTAL\_PT</errorId>

<exception>gov.va.med.cds.ars.exception.ValidationException</exception>

<exceptionMessage>CENSUS\_REPORT\_INVALID\_START\_DATE</exceptionMessage>

<errorCode>CENSUS\_REPORT\_INVALID\_START\_DATE</errorCode>

<displayMessage>Census report start date: 2005-10-08 is invalid</displayMessage>

</fatalErrors>

</errorSection>

</report:report>

#### VIM Writes

VIM clinical data Write clients submit Write requests to create or update clinical data to the Repositories data sources. The Write requests are audited and validated before being persisted to the HDR database.

##### CDS Write Request

A VIM Write request is used by the VIM Write clinical data client to store clinical data for a patient into the HDR DB. The input parameters for the Write request are as follows:

* The create request contains a VIM template XML document that represents the clinical data to be stored.
* The template ID specifies the VIM write template schema to be used to validate the VIM template XML document create request. Template IDs are defined for each supported clinical domain. These include OP, Allergy, Vitals, and Laboratory (Chemistry/Hematology).
* The request ID is unique to the sending site and is used to identify the Write request within the system.
* The optional clientName is used for auditing and monitoring the Create/Update requests. The clientName is limited to 100 characters. If the clientName is more than 100 characters, CDS shall return a validation exception to the clients.
* The optional clientRequestInitiationTime contains the time at which the Create/Update request is initiated at the client’s instance.

The string value returned by this request is an XML document that is an instance of the CDS response schema. This tells the requester whether CDS successfully stored the data into HDR and identifies any error conditions in the case of failure. Clinical data is not included in this instance, just status and error information (if any).

Write clients use VIM to store data in the HDR database. In the case of HTH, all HL7 Write requests are submitted to a domain-specific Java Messaging Service (JMS) queue via a socket adapter, which are transformed to VIM by the CDS Message Mediator component for persistence.

An Update request is identical to a Write request, except for an additional data element that contains the record identifier of the record being updated. An Update request is identical to a Write request, except for an additional data element that contains the record identifier of the record being updated.

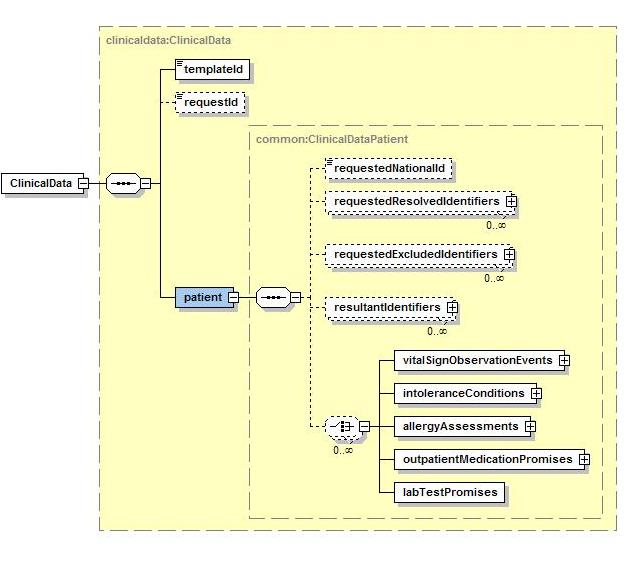
##### Write Request Response

When a Write (Create) is successful, the response that is sent has an empty error section. If issues occur during a Create that result in the record not being written into the HDR then a response is sent with an error section containing a detailed error and explanation. An example of the write or create response is illustrated in the Figure below.

##### Write Request Template

A Create or Update clinical data template is used by client applications to persist clinical records to the HDR DB. At the schematic view, they are nearly identical to a Read template, however, the meta data differs and the error section is missing (a separate response template is used for writes). Write templates define a single entry point – with the exception of HTH Surveys which can define two survey entry points. An example of a Write template is illustrated in the Figure below.

Figure 33 Write Request Template



#### HL7 Writes

The HL7 Write clients connect to either the CDS Message Mediator or CDS Socket Adapter when persisting clinical data to the HDR database. The HL7 message is transformed to VIM by the CDS Message Mediator prior to persistence to the HDR database. HL7 write clients requiring data to be persisted in VistA, connects to FPDS JMS queues.

* HL7 messages from VistA come through the CDS Message Mediator and include: Allergies, Outpatient Pharmacy, Laboratory Chemistry and Hematology (Lab can come through the Socket Adapter and the CDS Message Mediator), and Vitals
* HL7 messages from HTH come through the Socket Adapter and include: Vitals, multi-patient Census Records, Patient Satisfaction Surveys, DMPs and ADLs.
* Include FPDS writes to VistA – HL7 messages from FtP come through FPDS and include Notes.

## Interface Detailed Design

### HTH Client

The HDR 3.15 VIP Build 1 release includes enhancement to the HTH client to provide census write modifications, validation and handling of census activity reports and patient satisfaction survey titles.

Table 41 HTH Client

|  |  |
| --- | --- |
| System | Details |
| Title | Home Telehealth |
| Abbreviation | HTH |
| Point of Contact | George Blankenship / Chris Woodyard |
| Vendor | VA |

|  |
| --- |
| **HTH Reads** |
| The Read request submitted by HTH to CDS is a SOAP Web service Read request for HTH Census, Disease Management Protocol (DMP), Activities of Daily Living (ADL) and Patient Satisfaction Survey (PSS) data. CDS returns the Read response to HTH on the same connection. HDR supports synchronous Read requests from HTH to retrieve HTH Census, DMP, ADL and PSS aggregate values from HDR via the Aggregate Read Service (ARS). |
| **HTH Read Filters and Templates** |
| HTH uses the following XML schema combination for retrieval of HTH Survey, DMP and Census data from the HDR ARS.   * HTResponse.xsd * HTReportFilter.xsd * ReportFilter.xsd   The following filter schemas are used for a Read request between ARS and the HTH client application:   * HTReportFilter.xsd * ReportFilter.xsd |

### FtP Client

The HDR 3.15 VIP Build 1 release includes additional read functionality to return VistA User detail, including the IEN, for the FtP client and logging access to sensitive patients.

Table 42 FtP Client

| System | Details |
| --- | --- |
| Title | Fix the Phones |
| Abbreviation | FtP |
| Point of Contact | Dennis Peterson |
| Vendor | VA |

|  |
| --- |
| **FtP Reads / Writes** |
| FtP reads VistA data through the HDR services, such as CDS, Pathways and FPDS and writes to VistA via HL7. |
| **FtP Pathways Read Filters and Templates** |
| FtP uses various filters and templates. Refer to Table 32. |

# Human-Machine Interface

Not Applicable. HDR is a back end system only with no user interface.

## Interface Design Rules

Not Applicable.

## Inputs

Not Applicable.

## Outputs

Not Applicable.

## Navigation Hierarchy

Not Applicable.

1. Additional Information
   1. Identification of Technology and Standards

Refer to Table 21 – HDR TRM Technologies / Tools

* 1. Constraining Policies, Directives and Procedures

Design constraints are subject to the enterprise VA Technical Reference Model and Standards Profile (TRM/SP), which is a guide for the use of tools and programming languages. The TRM/SP includes guidance and policy regarding operating systems, database servers, and application servers. The PD ProPath VIP process provides guidelines for software development. Any variance from these guidelines shall be approved by appropriate waiver.

The VistA HL7 package cannot communicate to a Java JMS module and hence requests must be sent to the HL7 Listener using a socket connection. Commit Acks/Errors are returned on the initial connection and secondary Application Acks/Error responses are configured to be sent from VistA HL7 Listener through the HDR socket adapter to the appropriate response JMS Queue.

* 1. Requirements Traceability Matrix

HDR is transitioning from the Project Management Accountability System (PMAS) to Veteran-foused Integration Process (VIP) for the HDR 3.15, VIP Build 1 release, however, there are still some areas that are not fully defined yet. All requirements have been defined with traceability in CCM, RM and QM. However, authorized access is limited to those that are required to review and approve requirements traceability.

* 1. Packaging and Installation

Refer to the *HDR 3.15 Deployment, Installation, Back-out and Rollback Plan* that is posted in CCM using the link above. Deployment is to the AITC only as a national release.

* 1. Design Metrics

Refer to Table 3 – Workload and Performance Requirements

1. Sample Design Screen Shots
   1. Write Request Examples

Figure 34 Allergies Write Request Template

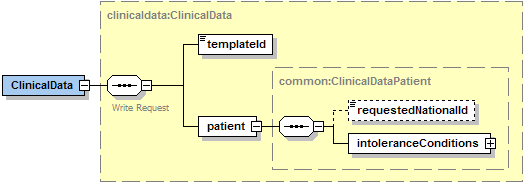


Figure 35 ADL Data Write Template

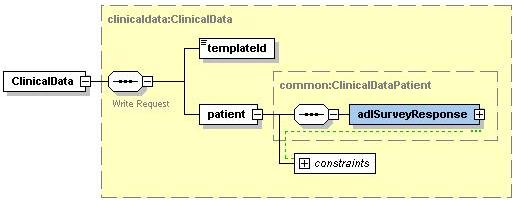


Figure 36 HTH DMP Response Write Template

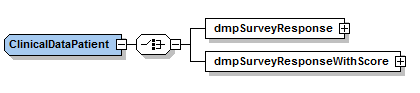
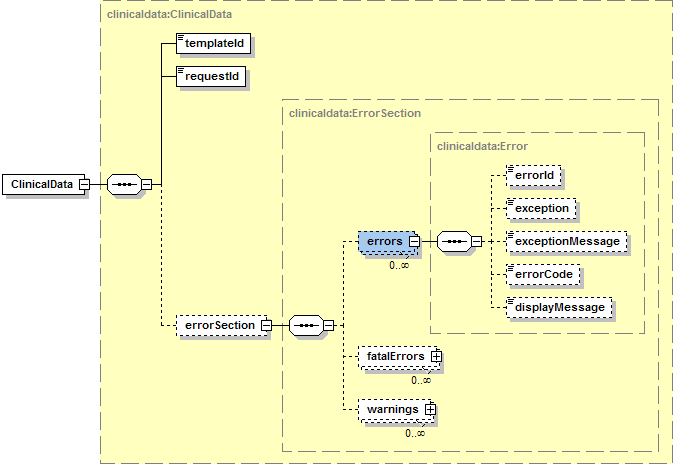


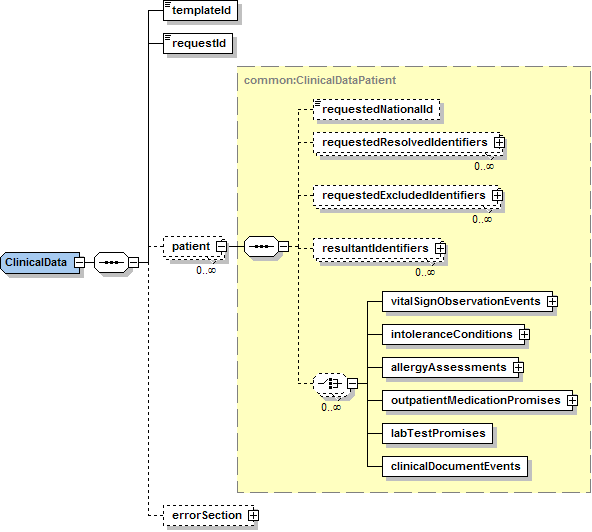
Figure 37 Write Clinical Data Response Template



* 1. Read Response Examples

The following is the general schematic of a read response – the examples following are the schematic of individual entry points such as vitalSignObservationEvents, intoleranceConditions, labTestPromises, etc.

Figure 38 Read Response Template



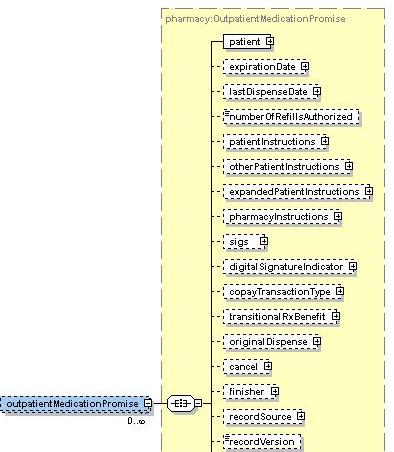


Figure 39 OP Read Response Template

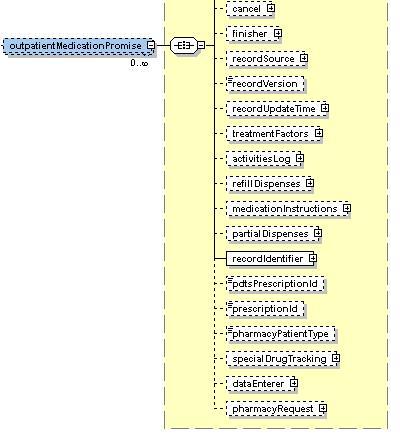


Figure 40 Lab CH Read Response Template

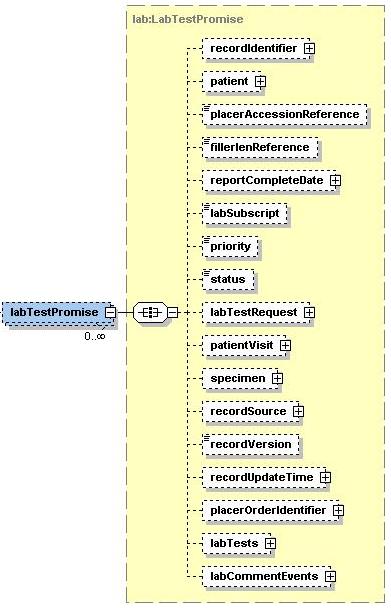


Figure 41 HTH Surveys Read Response Template

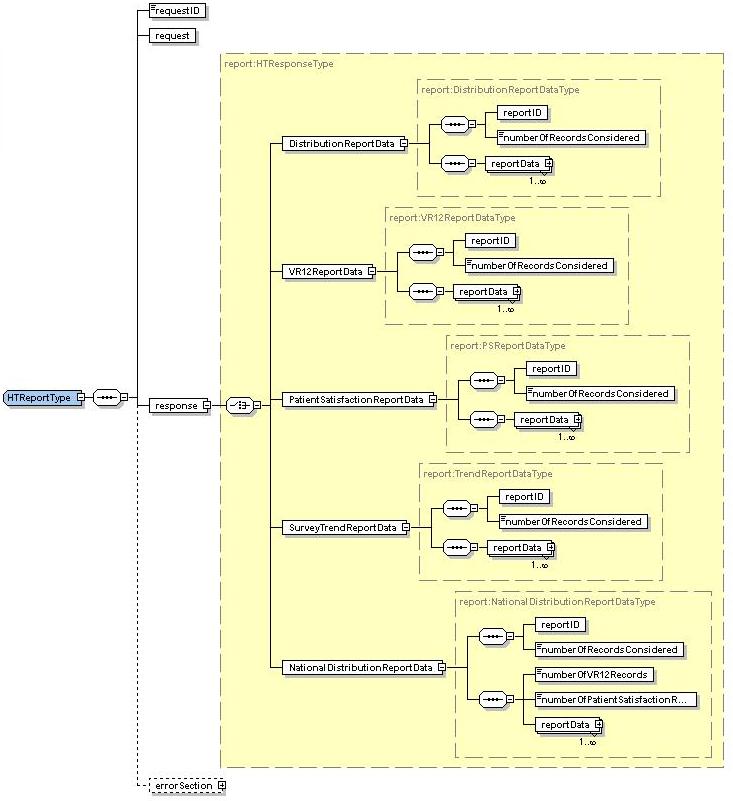


Figure 42 Non-VA Medications Read Response Template

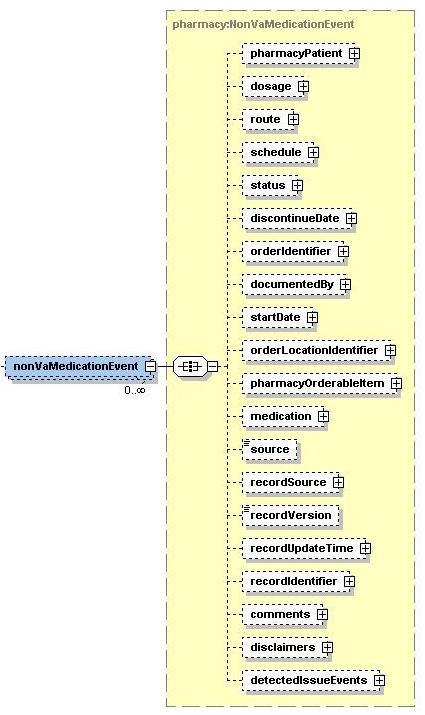


Figure 43 Immunization Read Response Template

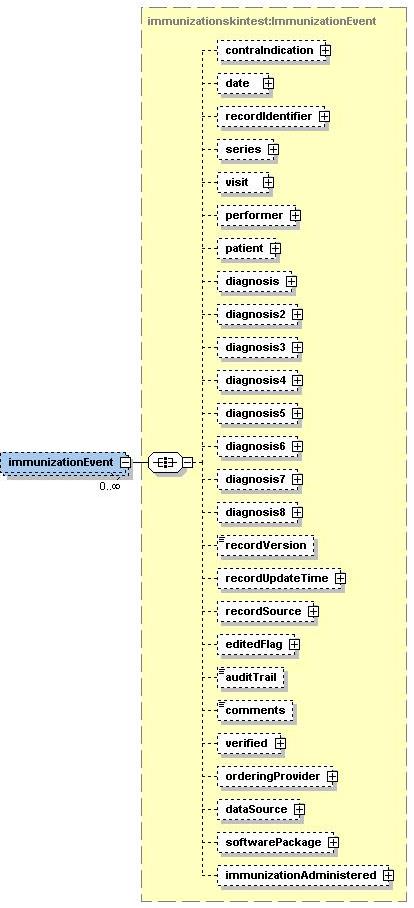


Figure 44 Skin Test Read Response Template

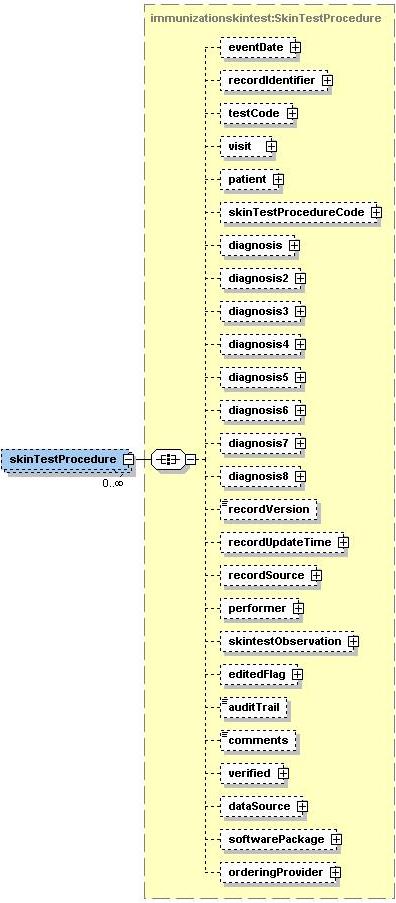


Figure 45 Problem List Read Response Template

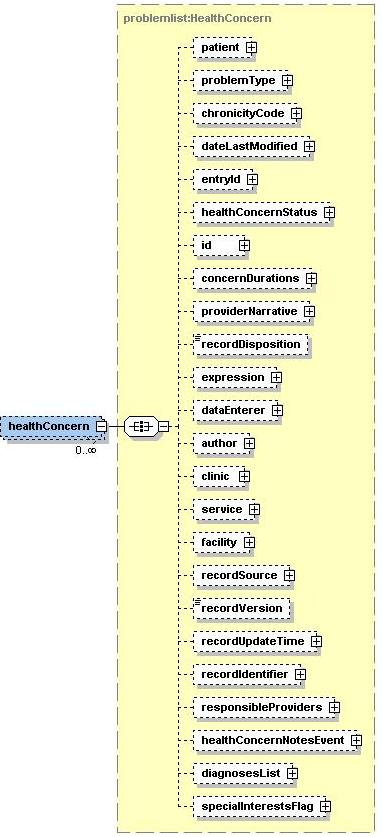


Figure 46 PSS Schematic

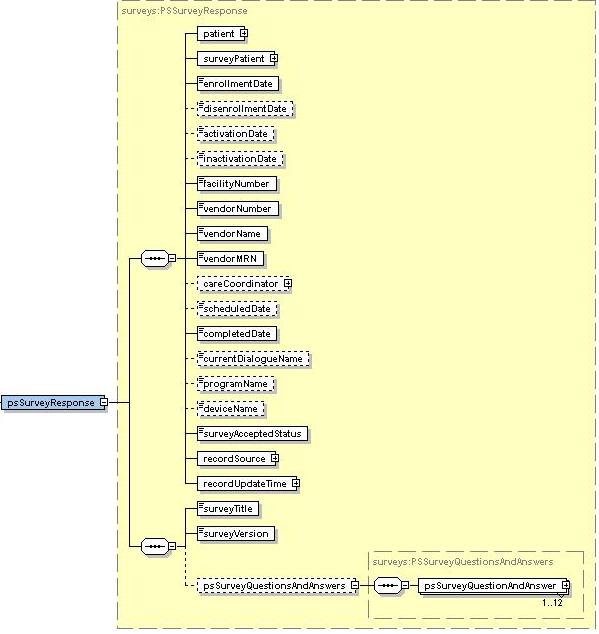


Figure 47 ADL Schematic

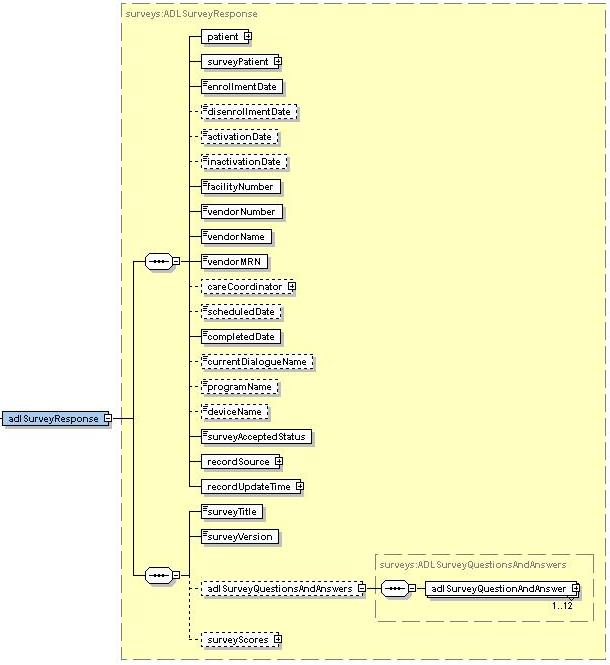


Figure 48 Appointment Read Response Template

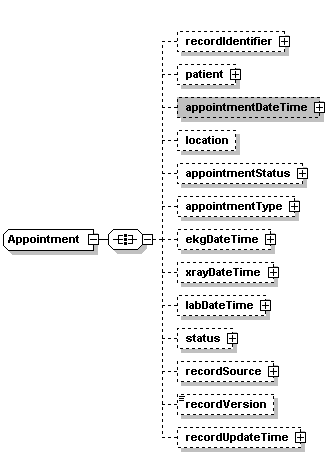


Figure 49 DMP Generic Response

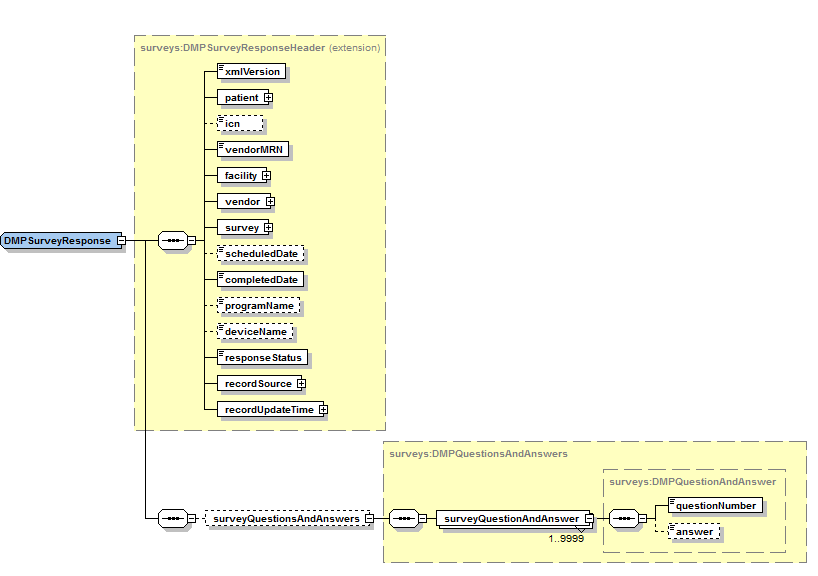
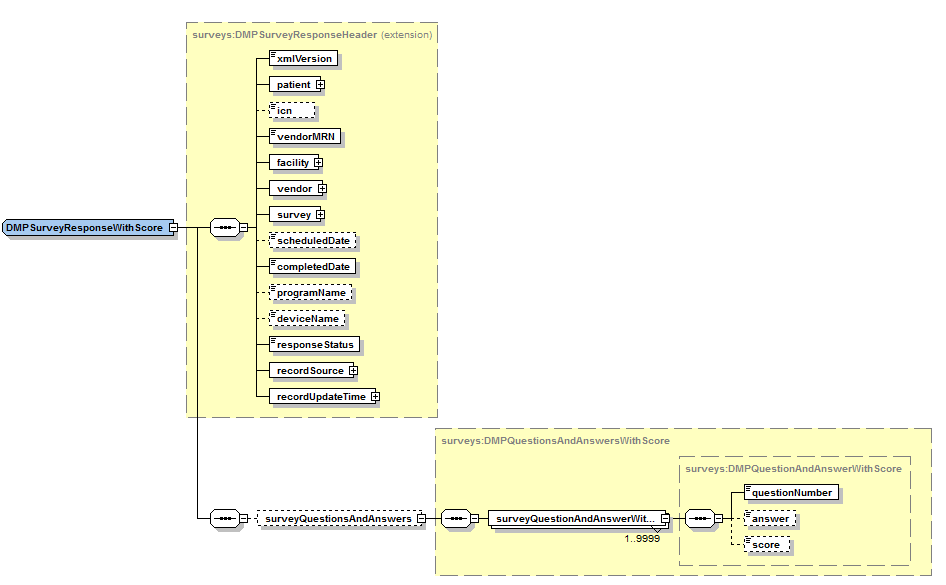
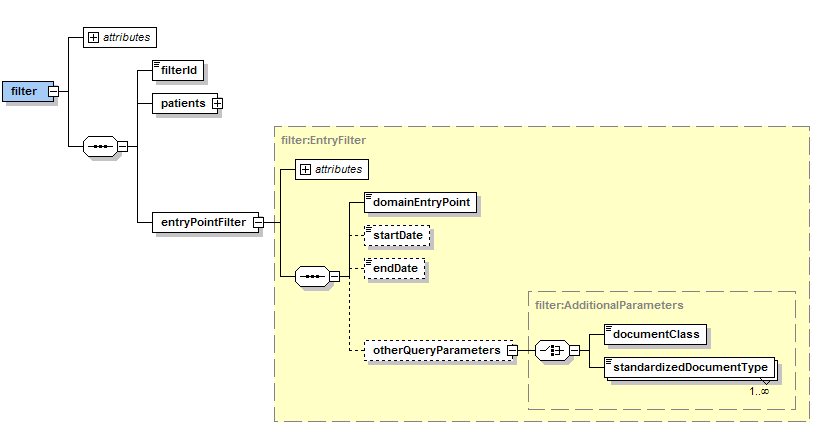


Figure 50 DMP Generic Response with Scores



* 1. Filter Examples

Figure 51 TIU Domain Read Filter



Note: Due to the hierarchical nature of the document class, and because a resolution of the sub-class to the parent class is necessary, querying TIU data by document class will certainly be less responsive than querying TIU data by standardized document type.

Figure 52 Non-VA Medications Filter schematic

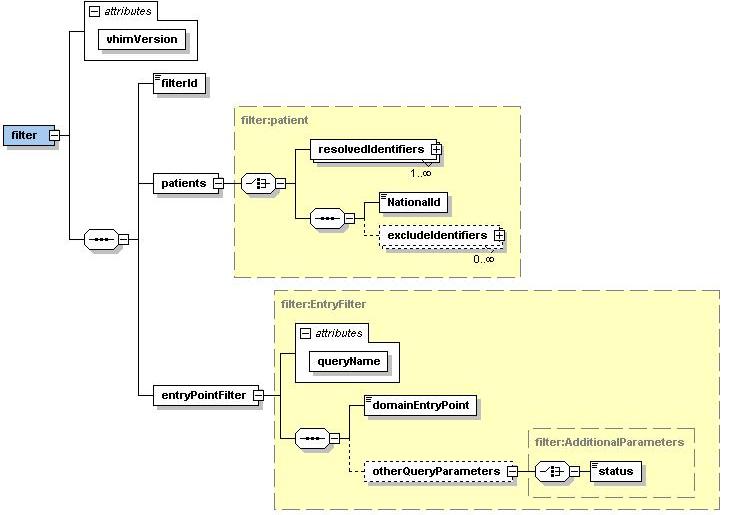


Figure 53 Immunizations Filter Schematic

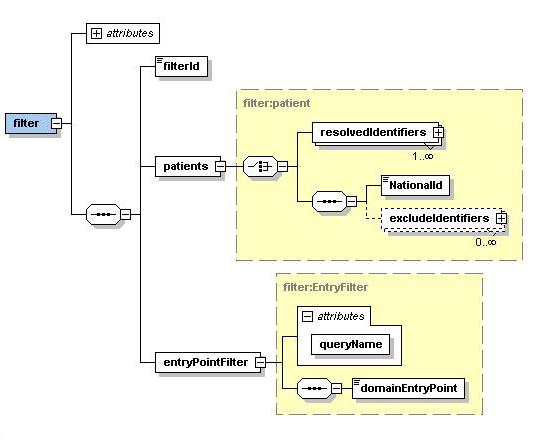


Figure 54 Skin Test Filter Schematic

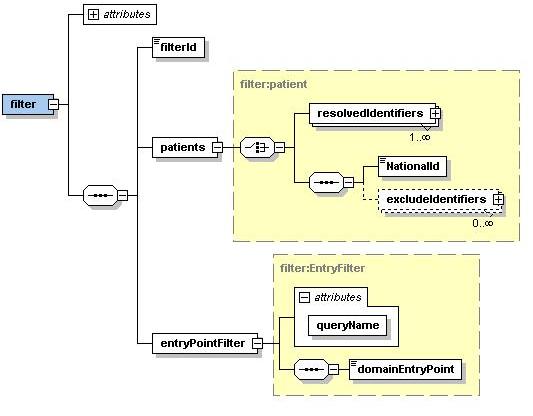


Figure 55 Problem List Filter Schematic

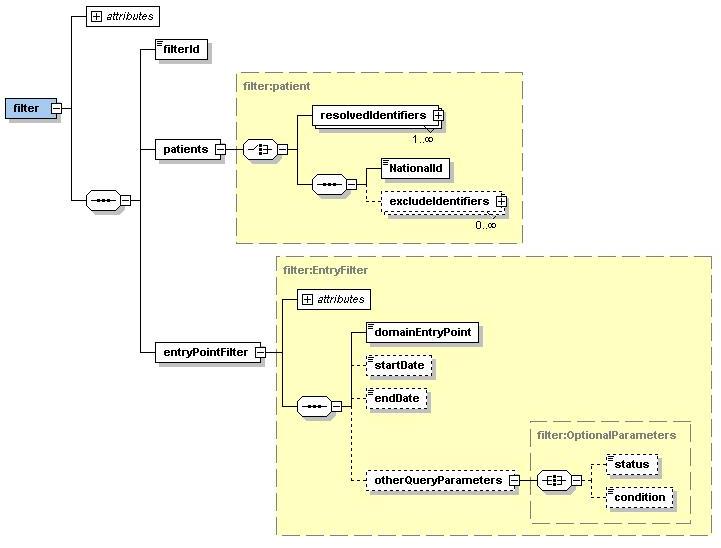


Figure 56 AppointmentSinglePatient Filter Schematic

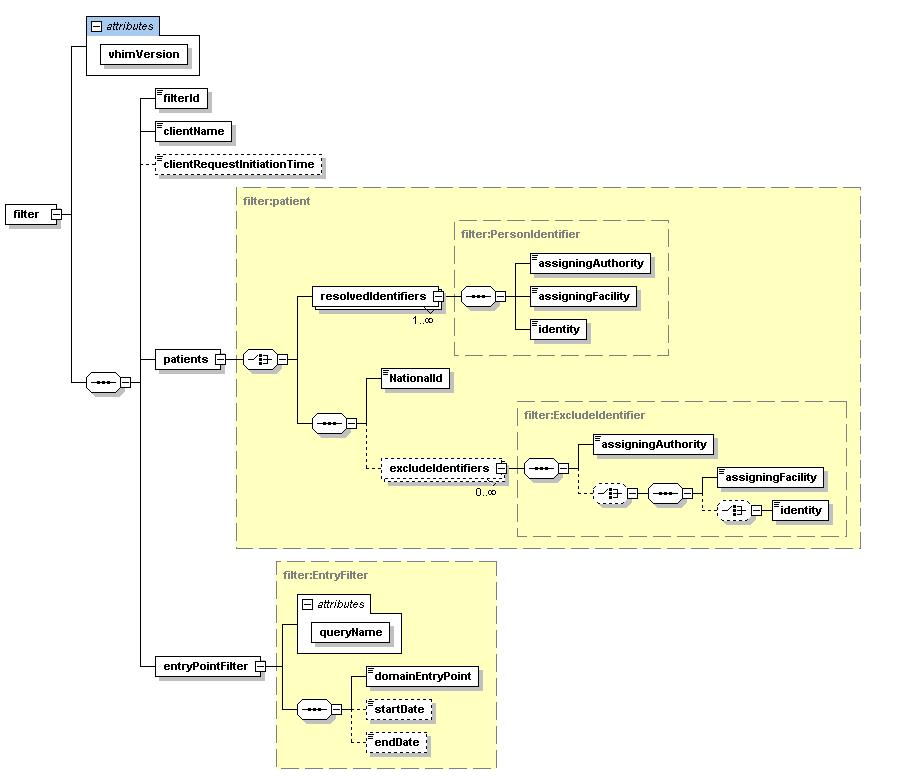


Figure 57 RequestsandExamsSinglePatient Filter Schematic

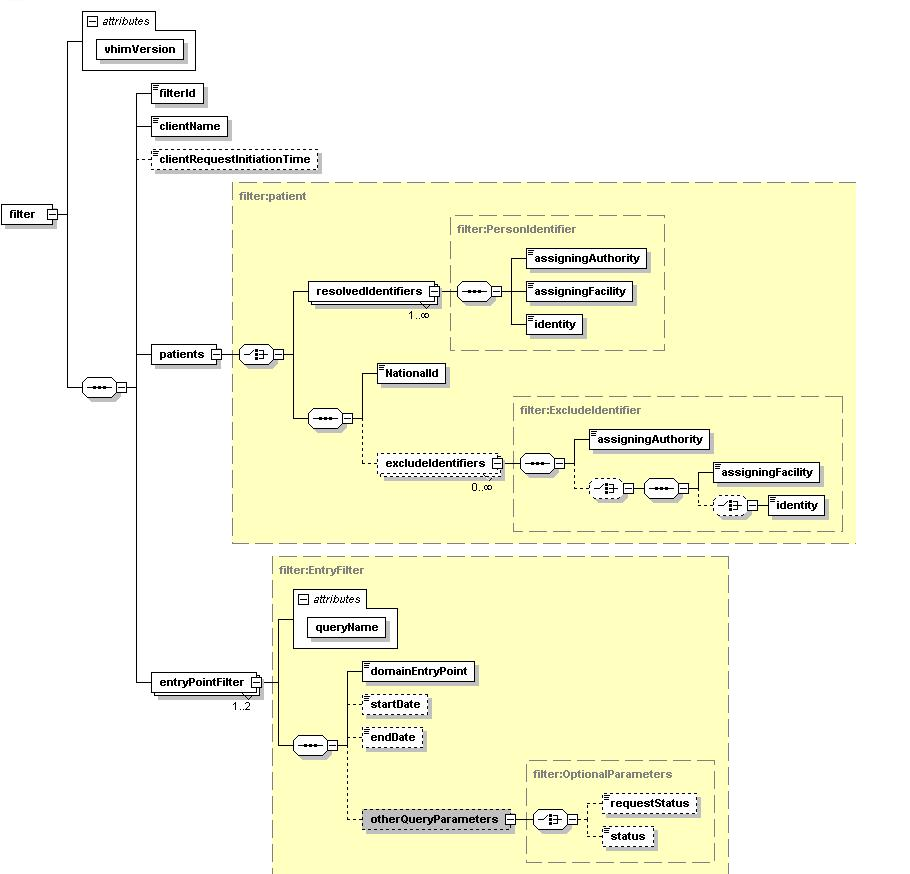


Figure 58 HTH Survey Reads Request Filter

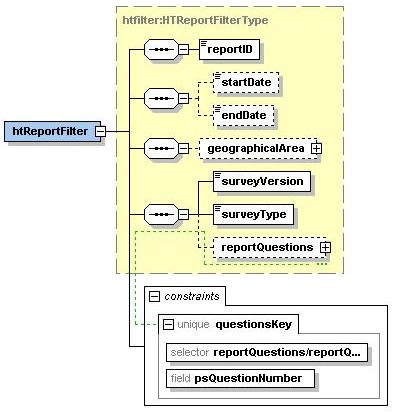


Figure 59 MHV Allergies Read Filter

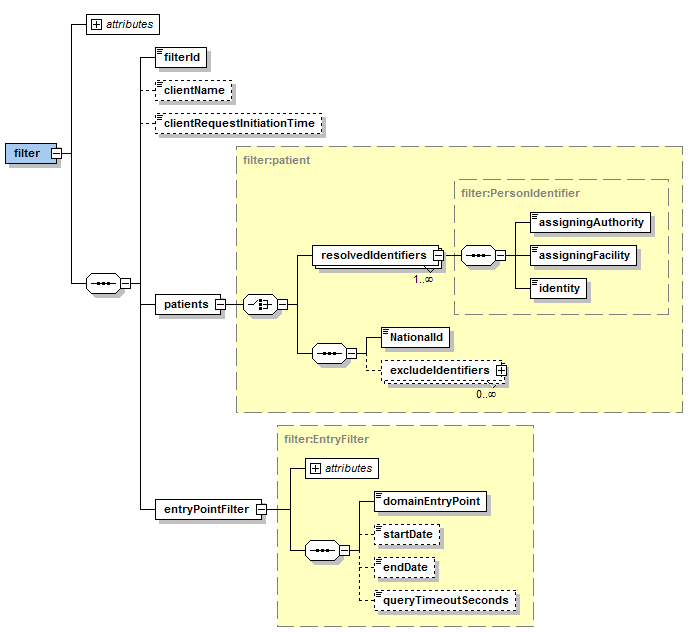


Figure 60 MHV Lab Read Filter

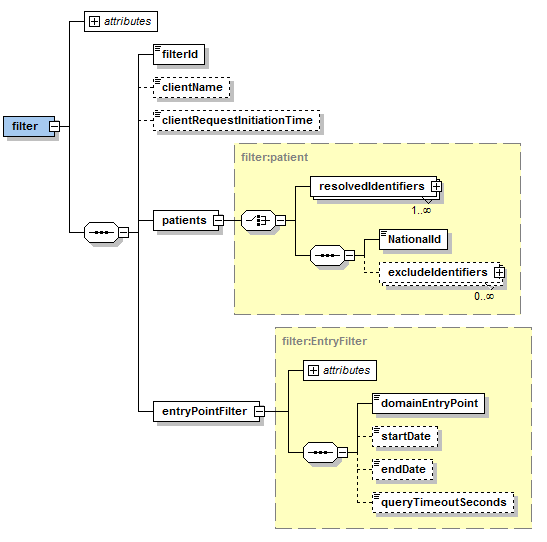


Figure 61 RDI Allergies and OP Reads Data Filter

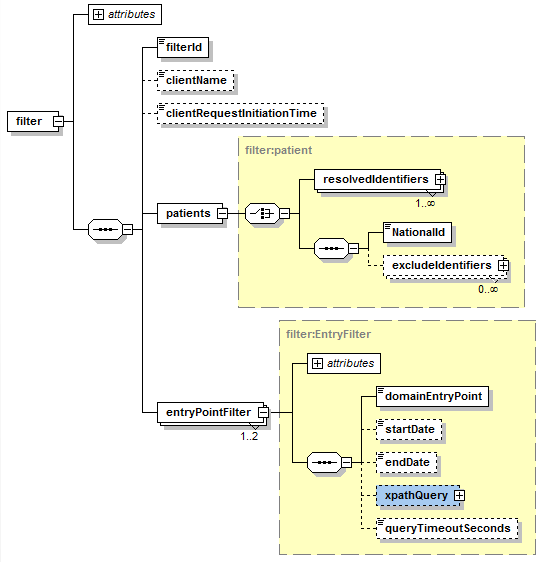
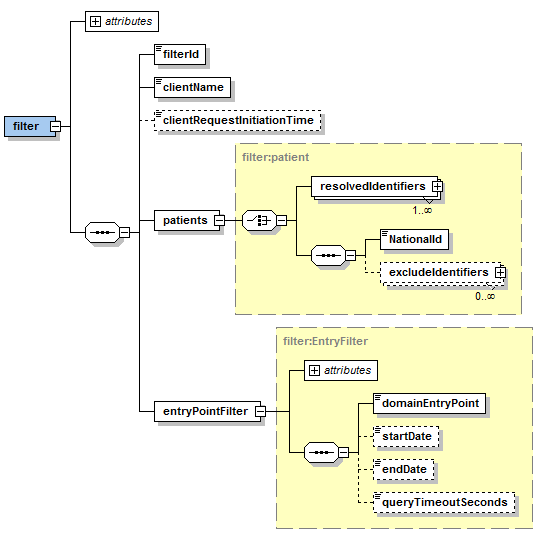
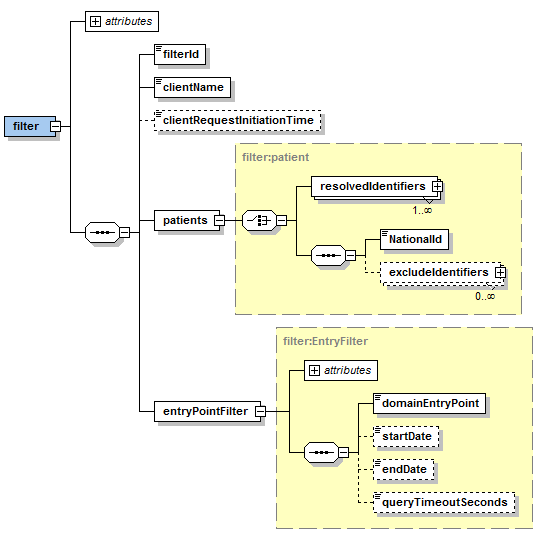
 

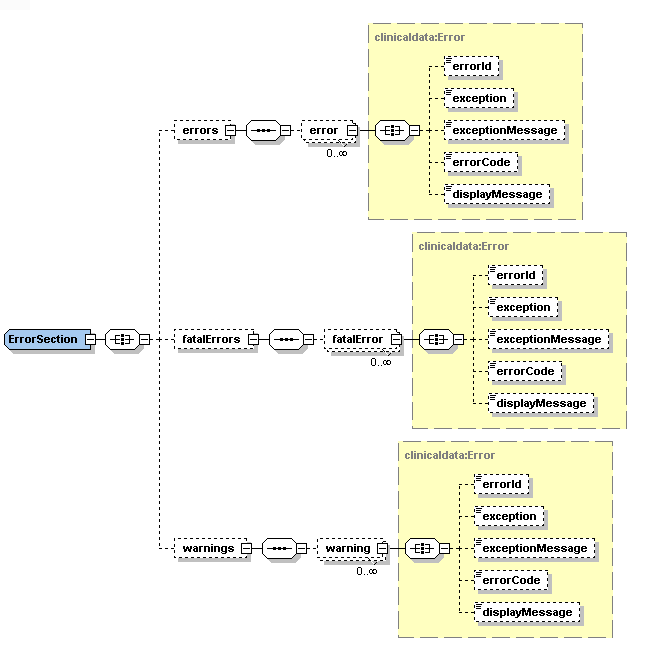
Figure 62 Mobile Health Appointments Read Data Filter



* 1. Error Messages

Processing errors that occur in the processing of a Read/Write request result in exceptions that generate internal error messages. These internal error messages are returned to the client application as part of the error section in the response message, detailed in the Figure below.

Figure 63 Error Section in Read request Response Message



The definitions of the fields for Error Type are as follows:

* **errorId**: an internally generated sequence value, not guaranteed to be unique, that links the error with internal CDS/Pathways log entries.
* **exception**: defines the general class of error (the Java class name in the CDS/Pathways exception framework).
* **exceptionMessage**: included for future scenarios in which it would be useful to capture stack-trace and external-dependency-generated messages. Currently, this value is usually null.
* **errorCode**: a computable identifier for the error. Currently, it is a string value. In the future it may be converted to another format.
* **displayMessage**: a parameterized text message, hopefully informative to the recipient.

The following tables lists Exceptions and Errors with possible errors that may be returned to a user in a response from HDR 3.15.

Table 43 Exceptions and Errors

| Exception | Error Code | Display Message |
| --- | --- | --- |
| Persistence Errors | HIBERNATE\_SESSION\_FAILURE | Unable to get the Hibernate session for the HDR database. |
| HIBERNATE\_READ\_FAILURE | Failed to read from the Hibernate layer. The error is {0}. |
| HIBERNATE\_DELETE\_FAILURE | Failed to delete using the Hibernate layer. |
| NON\_UNIQUE\_IDENTITY | Could not find unique record with this identity: {0}. The reason is {1}. |
| CANT\_CREATE\_IDENTITY | Failed to create an IdentityInterface object with this Identity: {0}. |
| UNABLE\_TO\_RESOLVE\_WRITEABLE\_PERSISTENCE\_MANAGER\_FAILURE | Unable to resolve writeable persistence manager applicable to template meta data. |
| HDRII\_OPERATION\_FAILED | {0} Error: {1}. Working with object: {2}. |
| HDRII\_CONSTRAINT\_VIOLATED | HDR Integrity constraint violated: {0}. |
| ROOT\_CAUSE\_MSG | CDS Persistence Failure Message: {1}. |
| Persistence Manager Exceptions | READ\_PERSISTENCE\_MGR\_NO\_PERSON\_WITH\_SITE\_ID | There is no person identifier in the list of person identifiers that has a site identifier of {0}. |
| INVALID\_OR\_UNEXPECTED\_QUERY\_PARAM | The query parameter '{0}' is not expected or is invalid. |
| READ\_PERSISTENCE\_MGR\_NULL\_PERSON\_IDENTIFIER\_LIST | The persistence manager{0}requires that the person identifiers list contains values |
| Model Assembler Exception | MODEL\_ASSEMBLER\_EXCEPTION\_NOT\_ALL\_RESULTS\_ASSEMBLED | Not all results were assembled. The domain is '{0}' and the association name is '{1}'. |
| Request Error | REQUEST\_TYPE\_MISMATCH\_OPERATION\_VS\_INSTANCE | The API request operation must match the request type defined in the instance XML. |
| CDS\_2X\_SERVICE\_FAILURE | Service Call failed on CDS 2xservice.The exception is{0} and the reason is{1} |
| Write Requests | WRITE\_REQUEST\_XML\_NULL | Write request XML cannot be null or empty. |
| WRITE\_REQUEST\_REQUEST\_ID\_NULL | Write request requestId cannot be null or empty. |
| WRITE\_REQUEST\_TEMPLATE\_ID\_INVALID | Invalid templateId {0}. The schema is not loaded in cache. The reason is {1} |
| WRITE\_REQUEST\_SIMPLE\_INVALID\_TEMPLATE\_ID  WRITE\_REQUEST\_TEMPLATE\_ID\_NULL | Invalid template identifier {0}.  Write request template Id cannot be null or empty. |
| Read Exceptions | UNEXPECTED\_READ\_EXCEPTION | An unexpected exception occurred while reading data for patient. Assigning Facility is {0}. Patient ID is {1}. Domain entry point is {2}. The message is {3} |
| READ\_REQUEST\_TEMPLATE\_ID\_NULL | Read request template id cannot be null or empty. |
| READ\_REQUEST\_FILTER\_ID\_NULL | Read request filter id cannot be null or empty. |
| READ\_REQUEST\_FILTER\_XML\_NULL | Read request filter XML cannot be null or empty. |
| READ\_REQUEST\_REQUEST\_ID\_NULL | Read request requestid cannot be null or empty. |
| READ\_REQUEST\_FILTER\_VHIM\_VERSION\_INVALID | Read request filter VHIM version is invalid {0}. |
| READ\_REQUEST\_TEMPLATE\_ID\_INVALID | Invalid templateId {0}. The schema is not loaded in cache. |
| REQUEST\_TYPE\_MISMATCH\_TEMPLATE\_VHIM\_VERSION\_FILTER\_VHIM\_VERSION | The requested Read template VIM version {0} must match the filter VHIM version {1}. |
| REQUEST\_TYPE\_MISMATCH\_TEMPLATE\_ENTRYPOINT\_FILTER\_ENTRYPOINT | The requested Read template entry points {0} must be a subset of the filter entry points {1}. |
| READ\_RESULT\_INCORRECTLY\_ACCESSED | The state of the ReadResult does not support this operation. |
| READ\_REQUEST\_DATA\_SOURCE\_FAILURE | Assigning Facility is {0}. Patient ID is {1}. Domain entry point is {2}. Database is {3}. The Error reason is {4}, , LoggingSeverity.WARNING. |
| READ\_REQUEST\_ALL\_DATASOURCES\_FAILED | All datasources failed. Unable to get data from applicable datasource(s) due to the following reason:\n {0}. |
| READ\_REQUEST\_WORK\_SCHEDULING\_EXCEPTION | Could not schedule work, LoggingSeverity.WARNING. |
| READ\_REQUEST\_WORK\_WAIT\_INTERRUPTED\_EXCEPTION | InterruptException occurred while waiting for scheduled work to finish, LoggingSeverity.WARNING. |
| CANT\_MATCH\_TEMPLATE\_TO\_DELEGATE | Cannot match template {0} to available service delegate. |
| READ\_REQUEST\_INPUT\_PARAMETERS\_NULL | Read request cannot have empty input parameters(s).Empty input parameter(s) are{0} |
| WRITE\_REQUEST\_INPUT\_PARAMETERS\_NULL | Write request cannot have empty input parameters(s). Empty input parameter(s) are{0} |
| READ\_QUERY\_STRATEGY\_NULL\_FAILURE | The Read query strategy for the template{0} and the domain entry point{2} is not defined and is expected to be LoggingSeverity.WARNING |
| INVALID\_ASSIGNING\_FACILITIES\_IN\_FILTER\_XML\_REQUEST | Invalid Assigning Facility mentioned in the Filter, LoggingSeverity.WARNING |
| ALL\_ASSIGNING\_FACILITIES\_INVALID\_IN\_FILTER\_XML\_REQUEST | All the Assigning Facilities mentioned in the Filter are invalid and they are {0}. |
| XML Parsing Exception | XSLT\_HELPER\_TRANSLATION\_EXCEPTION | An error occurred while transforming the data {0}. |
| XML Aggregation Exceptions | XML\_AGGREGATION\_ZERO\_INPUT\_SOURCES | XML aggregation requires at least one XML input source. |
| XML\_AGGREGATION\_VHIM\_VERSION\_INVALID | No XML Aggregator found for VIM version {0}. |
| XML\_AGGREGATION\_FAILED | XML aggregation failed. |
| Filter Errors | INVALID\_XPATH\_QUERY | Invalid xpath query. |
| ERROR\_MARSHALLING\_FILTER\_XML | Error unmarshalling filter XML. Filter XML was {1}. The reason is {0}. |
| NO\_ENTRYPOINTS\_IN\_FILTER | No entry points specified in filter: {0}. |
| NO\_PATIENT\_IDS\_REQUESTED | No patient identifiers are requested in the filter. |
| NO\_PATIENT\_IDS\_RESOLVED | No patient identifiers could be resolved from the filter |
| ALL\_PATIENT\_IDS\_EXCLUDED | All patient identifiers were excluded by the filter. |
| PATIENT\_ID\_MISSING\_ASSIGNING\_AUTHORITY | No assigning authority value in the patient identifier. |
| PATIENT\_ID\_MISSING\_ASSIGNING\_FACILITY | No assigning facility value in the patient identifier. |
| PATIENT\_ID\_MISSING\_IDENTITY | No identity value in the patient identifier. |
| CANNOT\_PARSE\_DATE | Cannot parse the date {0} specified in the filter. |
| Filter Patient Resolver Exceptions | FILTER\_PARSER\_DOM\_EXCEPTION | Unable to parse the request filter {0}. |
| MULTIPLE\_NATIONAL\_IDS\_IN\_FILTER | Multiple national identifiers specified in the filter {0} (only 1 permitted per patient). |
| ERROR\_OBTAINING\_CORRESPONDING\_IDS | Error while obtaining corresponding IDs for National Id {0}. |
| ERROR\_ADDING\_RESOLVEDIDENTIFIERS\_TO\_FILTER | Error adding resolved identifiers to filter. |
| Schema Helper Exceptions | SCHEMA\_VALIDATION\_FAILED | Schema validation failed because of {0}. |
| ERROR\_VALIDATIING\_XML\_AGAINST\_COMPILED\_SCHEMA | Error occurred when validating an XML instance against a compiled schema. The reason is {0},  LoggingSeverity.ERROR. |
| SCHEMA\_XML\_NULL | Schema XML document cannot be null. |
| FATAL\_ERROR\_VALIDATING\_XML\_AGAINST\_COMPILED\_SCHEMA | Fatal error occurred when validating an XML instance against a compiledSchema. Reason is {0} |
| Template Cache Exceptions | CANNOT\_CREATE\_TEMPLATE\_CACHE\_PERSISTENCE | Problem occurred creating template cache persistence: {0}. |
| CANNOT\_SAVE\_TEMPLATE | Problem occurred when persisting a template jar: {0}. The reason is {1}. |
| CANNOT\_LOAD\_TEMPLATE\_CACHE | Problem occurred when loading template cache. The reason is {0}. |
| CANNOT\_LOAD\_SCHEMA\_INTO\_TEMPLATE\_CACHE | Problem occurred when trying to load a schema into the template cache: {0}. The reason is {1}. |
| Template Meta Data Provider Exception | CANNOT\_LOAD\_TEMPLATE\_METADATA | Problem loading template meta data for templateId {0}. The reason is {1}. |
| SymbolMap Provider Exception | CANNOT\_CREATE\_TEMPLATE\_SCHEMA\_SYMBOL\_MAP | Problem creating the symbol map for a template schema: {0}. The reason is {1}. |
| TemplateService Exceptions | VTS\_SERVICE\_FAILED | The Template Service (VTS) failed: {0}. |
| CANNOT\_MANIPULATE\_TEMPLATE\_DATA | The data in the template cannot be manipulated correctly: {0}. |
| CANNOT\_LOAD\_TEMPLATE\_FROM\_VTS | Problem loading the template with templateId {0} from the VHIM Template Service. The reason is {1}. |
| NO\_TEMPLATE\_FOUND\_FROM\_VTS | Problem loading the template with templateId {0} from the VIM Template Service. No template returned. |
| Filter Cache Exceptions | CANNOT\_LOAD\_FILTER\_SCHEMA\_FROM\_TFS | Problem loading the filter with filterId {0} from filter service. The reason is {1}. |
| CANNOT\_LOAD\_FILTER\_SCHEMA\_FROM  \_PERSISTENCE | Problem loading the filter with filterId {0} from persistence. The reason is {1}. |
| CANNOT\_LOAD\_FILTER\_SCHEMA\_FROM\_FILESYSTEM | Problem loading the filter with filterId {0} from file system. The reason is {1}. |
| FILTER\_MEMORY\_CACHE\_IS\_NOT\_PROVIDED | The filterMemoryCache passed in is null. |
| CANNOT\_SAVE\_FILTER\_SCHEMA | Problem occurred when persisting a filter schema: {0}. The reason is {1}. |
| CANNOT\_FIND\_FILTER\_SCHEMA | Filter schema not found: {0}. |
| INVALID\_FILTER\_SCHEMA | Invalid filterId {0}. The filter schema is NOT loaded in cache. The reason is {1}. |
| UNABLE\_TO\_READ\_FILTER\_SCHEMA | Unable to Read filter schema with filterId {0}. |
| CANNOT\_FIND\_FILTER\_VALIDATOR | Filter schema validator not found in filter cache for filterId {0}. |
| CANNOT\_DELETE\_FILTER | Problem occurred when deleting a filter schema from database: {0}. The reason is {1}. |
| ERROR\_DELETING\_FILTER\_DOMAIN\_TABLES | Problem deleting filter domain tables: {0}. The reason is {1}. |
| Entity Resolver Exceptions | CANNOT\_GET\_STRING\_DATA | The entity resolver had a problem while getting the string data. |
| OPERATION\_NOT\_SUPPORTED | The method {0} is not supported. |
| MISSING\_SCHEMA\_JAR | Schema jar cannot be null. {0}. |
| UNABLE\_TO\_LOAD\_SCHEMAS\_INTO\_CACHE | Unable to load schemas into the cache for Template name {0}. Reason is {1}. |
| UNABLE\_TO\_LOAD\_SCHEMAS\_INTO\_RESOURCE | Unable to load schema into the resource. {0}. The reason is {1}. |
| INVALID\_TEMPLATE | Invalid templateId {0}. The schema is NOT loaded in cache. The reason is {1}. |
| IdM Exceptions | IDM\_NULL\_PERSON\_IDENTIFIER\_EXCEPTION | Person identifier cannot be null. |
| IDM\_NULL\_PERSON\_IDENTITY\_EXCEPTION | Person identity cannot be null. |
| IDM\_RESPONSE\_PROCESSING\_EXCEPTION | An error occurred while processing the IDM service response. |
| IDM\_ERROR\_RESPONSE | {0}. |
| Exception Handler Exceptions | TEMPLATE\_ID\_NULL | The <templateId> was null. |
| TEMPLATE\_HELPER\_CONFIGURATION\_EXCEPTION | TemplateHelper was not configured for class {0}. |
| MISSING\_ERROR\_SECTION\_HELPER\_EXCEPTION | No ErrorSectionHelperInterface was defined in the configuration for templateId {0}. |
| MISSING\_CLINICAL\_DATA\_RESPONSE\_EXCEPTION | No ClinicalDataResponseInterface was set while configuring class: {0}. |
| GUARANTEED\_LOGGER\_NOT\_CONFIGURED\_AND\_NULL | Guaranteed logger is not configured and is null. |
|  |  |
|  |  |
| Socket Client Timeout Exception | SOCKET\_CLIENT\_IO\_EXCEPTION | An IO exception has occurred. |
| SOCKET\_CLIENT\_ TIMEOUT\_EXCEPTION | A socket timeout exception occurred. |
| MOM Exception | MDB\_ON\_MESSAGE\_JMS\_EXCEPTION | Error processing message in MDBs onMessage: {0}. |
| MLLP Exception | MLLP\_ILLEGAL\_ENCODING\_EXCEPTION | MLLP illegal encoding. |
| Encryption Exceptions | ENCRYPTION\_MISSING\_KEYSTORE\_PASSWORD\_EXCEPTION | Missing KeyStore password in System Parameters (required for encryption and decryption). |
| ENCRYPTION\_MISSING\_KEYSTORE\_FILEPATH\_EXCEPTION | Missing KeyStore file path (required for encryption and decryption). |
| ENCRYPTION\_EXCEPTION | Encryption exception. |
| Audit CLOB Store Errors | CDS\_AUDIT\_CLOB\_STORE\_ERROR | Error in auditing VIM Write request. |
| CDS\_AUDIT\_LOG\_ERROR | Error in auditing VIM Read request. |

* 1. eHMP REST web-service read request detailed examples

Given below is the REST web-service read request URL format supported by CDS for eHMP. REST web-service URL supports HTTP GET requests and support multiple path and query parameters. Path parameters are separated from query parameters by ‘?’. Path parameters are required. The path parameters are separated by ‘/’. The query parameters are name value pairs and are optional except for the clientName value. Query parameters are separated by ‘&’.

Example of REST web-service read request URL format that defaults the response to JSON format:

http://<host>:<port>/repositories.med.DNS /fpds/<domain>?/templateId=<templateId>&filterId=<filterId>&nationalId=<patientId>&clientName=<clientName

Example of REST web-service read request URL format that needs the read response in VIM XML format:

http://<host>:<port>/repositories.med.DNS /fpds//<templateId>/<filterId>/<patientId>/<domain>?clientName=<clientName>&excludedIdentifier=<excluded Id>&siteId=<excluded Id site>&text=<text>&start=<yyyy-MM-dd>&stop=<yyyy-MM-dd>&max=<max> &id=<id>&requestId=<requestId>&\_type=xml

Path elements:

domain: The clinical or non-clinical domain identifier

Query parameters:

**Required:**

templateId: The ID of the template that CDS uses internally to represent the response format

filterId: The ID of the filter that CDS uses internally to represent the client specific request

patientId: The patient’s ICN

clientName – name of the client requesting the data from VistA systems.

requestId: A unique alpha-numeric string assigned to the request for audit purposes

\_type: The type of the read response format. Default is JSON. Specify this query parameter only when the read response format is needed in XML format.

**Optional query parameters:**

excludedIdentifier: This works in combination of ICN.The patient ID at the site to exclude data from the response.

resolvedIdentifier: Used when ICN is not used.

**Optional VPR filter query parameters:**

text: Boolean to include document text

start: The start date of search in yyyy-MM-dd format

end: The end date of search in yyyy-MM-dd format

max: The maximum number of items to return

id: The ID of the single record to return

**Optional CDS filter query parameters:**

The filter element name is dependent on the domain. Example of filtering Problem domain data based on status being active:

@.status=Active

Figure 64 JSON response with error section{"sites": [

{

"apiVersion": "1.01",

"params": {

"domain": " SLC ",

"systemId": "D"

},

"data": {

"totalItems": 0,

"items": []

}

},

{

"apiVersion": "1.01",

"params": {

"domain": "SLC",

"systemId": "1"

},

"data": {

"updated": "20140130231928",

"totalItems": 2,

"items": [

{

"displayName": "BP",

"facilityCode": 500,

"facilityName": "R",

"high": "210/110",

"kind": "Vital Sign",

"localId": 4,

"locationName": "2TE",

"locationUid": "u",

"low": "100/60",

"observed": 200304041518,

"result": "138/72",

"resulted": 20030404151847,

"summary": "BLOOD PRESSURE 138/72 mm[Hg]",

"typeCode": "u",

"typeName": "BLOOD PRESSURE",

"uid": "u",

"units": "mm[Hg]"

},

{

"displayName": "P",

"facilityCode": 0,

"facilityName": "R",

"high": 120,

"kind": "Vital Sign",

"localId": 2507,

"locationName": "20 MINUTE",

"locationUid": "u",

"low": 60,

"observed": 200304041518,

"result": 72,

"resulted": 20030404151847,

"summary": "PULSE 72 /min",

"typeCode": "u",

"typeName": "PULSE",

"uid": "u",

"units": "/min"

}] }}]}

Figure 65 XML format read response

<?xml version="1.0" encoding="UTF-8"?>

<clinicaldata:ClinicalData xmlns:clinicaldata="Clinicaldata">

<templateId>GenericObservationRead1</templateId>

<requestId>6</requestId>-<patients>-<patient>

<requestedNationalId>1</requestedNationalId>-<resultantIdentifiers>-<resultantIdentifier>

<identity>local\_id\_1</identity>

<assigningFacility>5</assigningFacility>

<assigningAuthority>USVHA</assigningAuthority>

</resultantIdentifier>-<resultantIdentifier>

<identity>local\_id\_2</identity>

<assigningFacility>6</assigningFacility>

<assigningAuthority>USVHA</assigningAuthority>

</resultantIdentifier>-<resultantIdentifier>

<identity>110</identity>

<assigningFacility>5</assigningFacility>

<assigningAuthority>USVHA</assigningAuthority>

</resultantIdentifier>-<resultantIdentifier>

<identity>1</identity>

<assigningFacility>2</assigningFacility>

<assigningAuthority>USVHA</assigningAuthority>

</resultantIdentifier>-<resultantIdentifier>

<identity>174</identity>

<assigningFacility>2</assigningFacility>

<assigningAuthority>USVHA</assigningAuthority>

</resultantIdentifier>

</resultantIdentifiers>-<genericObservations>-<genericObservation>-<recordIdentifier>

<identity>174</identity>

<namespaceId>2</namespaceId>

</recordIdentifier>-<observationValue>[{"apiVersion":"1.01","params":{"domain":"SLC","systemId":"D"},"data":{"totalItems":0,"items":[]}}]</observationValue>

</genericObservation>-<genericObservation>-<recordIdentifier>

<identity>1</identity>

<namespaceId>2</namespaceId>

</recordIdentifier>-<observationValue>[{"apiVersion":"1.01","params":{"domain":"SLC","systemId":"1"},"data":{"updated":"20140205105840","totalItems":14,"items":[{"displayName":"BP","facilityCode":500,"facilityName":"CAMP MASTER","high":"210\/110","kind":"Vital Sign","localId":2504,"locationName":"20 MINUTE","location":"","low":"100\/60","observed":200304041518,"result":"138\/72","resulted":20030404151847,"summary":"BLOOD PRESSURE 138\/72 mm[Hg]","typeCode":":4500634","typeName":"BLOOD PRESSURE","units":"\/min"}]}}]></observationValue>

</genericObservation>

</genericObservations>

</patient>

</patients>

</clinicaldata:ClinicalData>

Figure 66 JSON response with error section{"sites":

                   [{ "errorSection":{

                             "fatalErrors":{

                                      "fatalError":{

                   "exceptionMessage":"FPDS\_ERROR",

                   "displayMessage":"Response type can only be XML or JSON",

                    "exception":"gov.va.med.repositories.fpds.validator.DefaultReadRequestValidator$InvalidParameterValidationError",

                   "errorCode":"FPDS\_ERROR",

                   "errorId":"65756756"

          }}}}

          ]}

 {"sites": [{"errorSection": {"fatalErrors": {"fatalError": {

   "exceptionMessage": "IDM\_ERROR\_RESPONSE",

   "displayMessage": "IDM service error for the national id x. The IDM error response is <IDM\_RESPONSE type='ERROR' requestType='GET\_CORRESPONDING\_IDS'><RESULT type='AR'><ERROR code='GCID01'><TEXT><![CDATA[ICN/VPID Does Not Exist:gov.va.med.person.idmgmt.server.profile.Identifier@58ccee7f[NI,IDM,x,200M,USVHA,,,,]]]><\/TEXT><\/ERROR><\/RESULT><\/IDM\_RESPONSE>.",

   "exception": "gov.va.med.cds.exception.IdmPersonServiceException",

   "errorCode": "IDM\_ERROR\_RESPONSE",

   "errorId": "65756756"

}}}}]}

Figure 67 XML response with error section

<?xml version="1.0" encoding="UTF-8"?>

<clinicaldata:ClinicalData xmlns:clinicaldata="Clinicaldata">

<templateId>GenericObservationRead1</templateId>

<requestId>65756756</requestId> -<patients> -<patient>

<requestedResolvedIdentifiers/> -<resultantIdentifiers> -<resultantIdentifier>

<identity>xxxxxxxxxxx</identity>

<assigningFacility>xxx</assigningFacility>

<assigningAuthority>USVHA</assigningAuthority>

</resultantIdentifier>

</resultantIdentifiers>

<genericObservations/>

</patient>

</patients> -<errorSection> -<fatalErrors> -<fatalError>

<errorId>65756756</errorId>

<exception>gov.va.med.repositories.fpds.validator.DefaultReadRequestValidator$InvalidParameterValidationError</exception>

<exceptionMessage>FPDS\_ERROR</exceptionMessage>

<errorCode>FPDS\_ERROR</errorCode>

<displayMessage>Request parameter clientName was not provided or was empty.</displayMessage>

</fatalError>

</fatalErrors>

<errors> </errors>

<warnings> </warnings>

</errorSection>

</clinicaldata:ClinicalData>

* 1. HTH Census HL7 write request

MSH^~|\&^HTAPPL^200T6~XX.HL7.X.X.GOV~DNS^HT CENSUS^688HT~ XX.HL7.X.X.GOV~DNS^20130923080000-0600^^1M~T02^Census\_T\_2013-09-21\_IM-T^P^2.4^^^AL^AL^USA

OBX^1^TX^^^<HTCensus><Version>2</Version><Segment><Current>1</Current><Last>1</Last></Segment><VendorFacilityNumber>1X</VendorFacilityNumber><VendorName>Telehealth Emulator</VendorName><SubmissionDateTime>2006-01-01T00:00:15</SubmissionDateTime><ReportStartDate>2000-12-25</ReportStartDate><ReportEndDate>2000-12-31</ReportEndDate><Patient><SSN></SSN><ICN></ICN><VendorMRN>1</VendorMRN><HomeDevice><Model>Home Device</Model><Serial>2</Serial></HomeDevice><Name><Given></Given><Middle>M</Middle><Family>Patient</Family></Name><DOB></DOB><LevelOfCare>1</LevelOfCare><EnrollmentDate>2000-12-15</EnrollmentDate><DisenrollmentDate></DisenrollmentDate><ActivationDate>2000-12-15</ActivationDate><InactivationDate></InactivationDate><FacilityLocation></FacilityLocation><Compliance>.85</Compliance><Responses>6</Responses><Expectations>7</Expectations><DataAge>1</DataAge><CareCoordinator><DUZ>45873</DUZ><FacilityNumber>673</FacilityNumber><Name><Given>Eileen</Given><Middle></Middle><Family>Carenurse</Family></Name></CareCoordinator><ProgramName></ProgramName><DMP>Other-CHF Diabetes Weight</DMP><Measurement>Weight</Measurement><Measurement>Blood Pressure</Measurement><Measurement>Pulse</Measurement><Measurement>Blood Glucose</Measurement><Telecom>Internet</Telecom></Patient></HTCensus>

Sample Census HL7 Write request response

MSH^~|\&^HDR HTPS^200HD~DNS SVWLS01.AAC.DNS ~DNS^HTAPPL^200T3~SENDING\_DNS~DNS^20140203092019.984-0700^^ACK^00000000000000369344^P^2.4^^^AL^NE

Sample Census HL7 Write response with error section

MSH^~|\&^HDR HTPS^200HD~DNS SVWLS01.AAC.DNS ~DNS^HTAPPL^200T3~SENDING\_DNS~DNS^20140203092019.984-0700^^ACK^00000000000000369344^P^2.4^^^AL^NE MSA^AR^feb32014409120011123   
ERR^feb32014409120011123gov.va.med.cds.xml.schema.SchemaValidationExceptionSCHEMA\_VALIDATION\_FAILEDSCHEMA\_VALIDATION\_FAILEDSchema validation failed because of cvc-complex-type.2.4.a: Invalid content was found starting with element 'surveyAcceptedStatus'. One of '{responseStatus}' is expected.: cvc-complex-type.2.4.a: Invalid content was found starting with element 'surveyAcceptedStatus'. One of '{responseStatus}' is expected.^

* 1. FtP CDS Read Request

FtP CDS HTH Vitals and Vital sign type read request

<?xml version="1.0" encoding="UTF-8"?>

<filter:filter xmlns:filter="Filter" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" vhimVersion="Vhim\_4\_00">

<filterId>VITAL\_SINGLE\_PATIENT\_ALL\_DATA\_FILTER</filterId>

<clientName>FtP</clientName>

<clientRequestInitiationTime>2014-05-30T09:30:10.5</clientRequestInitiationTime>

<patients>

<NationalId>1</NationalId>

</patients>

<entryPointFilter queryName="A">

<domainEntryPoint>VitalSignObservationEvent</domainEntryPoint>

<startDate>2000-01-10</startDate>

<endDate>2015-08-13</endDate>

<xpathQuery>

<xpath>vitalSignObservationEvents[((recordSource/namespaceId[starts-with(., "200T")]) and (status != 'E'))]</xpath>

</xpathQuery>

<otherQueryParameters>

<vitalSignType>TEMPERATURE</vitalSignType>

</otherQueryParameters>

<queryTimeoutSeconds>600</queryTimeoutSeconds>

</entryPointFilter>

</filter:filter>

**FtP CDS Vista Vitals and Vital sign type read request:**

<?xml version="1.0" encoding="UTF-8"?>

<filter:filter xmlns:filter="Filter" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" vhimVersion="Vhim\_4\_00">

<filterId>VITAL\_SINGLE\_PATIENT\_ALL\_DATA\_FILTER</filterId>

<clientName>FtP</clientName>

<clientRequestInitiationTime>2014-05-30T09:30:10.5</clientRequestInitiationTime>

<patients>

<NationalId>1</NationalId>

</patients>

<entryPointFilter queryName="A">

<domainEntryPoint>VitalSignObservationEvent</domainEntryPoint>

<startDate>2000-01-10</startDate>

<endDate>2015-08-13</endDate>

<xpathQuery>

<xpath>vitalSignObservationEvents[((recordSource/namespaceId[not(starts-with(.,'200T'))]) and (status != 'E'))]</xpath>

</xpathQuery>

<otherQueryParameters>

<vitalSignType>TEMPERATURE</vitalSignType>

</otherQueryParameters>

<queryTimeoutSeconds>600</queryTimeoutSeconds>

</entryPointFilter>

</filter:filter>

**FtP CDS Vista and HTH Vitals and Vital sign type read request:**

<?xml version="1.0" encoding="UTF-8"?>

<filter:filter xmlns:filter="Filter" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" vhimVersion="Vhim\_4\_00">

<filterId>VITAL\_SINGLE\_PATIENT\_ALL\_DATA\_FILTER</filterId>

<clientName>FtP</clientName>

<clientRequestInitiationTime>2014-05-30T09:30:10.5</clientRequestInitiationTime>

<patients>

<NationalId>1</NationalId>

</patients>

<entryPointFilter queryName="A">

<domainEntryPoint>VitalSignObservationEvent</domainEntryPoint>

<startDate>2000-01-10</startDate>

<endDate>2015-08-13</endDate>

<xpathQuery>

<xpath>vitalSignObservationEvents[(status != 'E')]</xpath>

</xpathQuery>

<otherQueryParameters>

<vitalSignType>TEMPERATURE</vitalSignType>

</otherQueryParameters>

<queryTimeoutSeconds>600</queryTimeoutSeconds>

</entryPointFilter>

</filter:filter>

* 1. FtP FPDS Read with CDS Filters

**Order domain:**

CDS filters by provider or/and service.

http://<host>:<port>/repositories.med.DNS /fpds/<domain>?/templateId=<templateId>&filterId=<filterId>&nationalId=<patientId>&clientName=<client

**Nonvamed domain:**

CDS filters by type, medication name or/and status.

http://<host>:<port>/repositories.med.DNS /fpds//<templateId>/<filterId>/<patientId>/<domain>?clientName=<clientName>&excludedIdentifier=<excluded Id>&siteId=<excluded Id site>&text=<text>&start=<yyyy-MM-dd>&stop=<yyyy-MM-dd>&max=<max> &id=<id>&requestId=<requestId>&\_type=json**&@.type=OTC&@.name =<name>&@.medStatusName=<status>**

**Lab domain:**

CDS filters by lab test name.

http://<host>:<port>/repositories.med.DNS /fpds//<templateId>/<filterId>/<patientId>/<domain>?clientName=<clientName>&excludedIdentifier=<excluded Id>&siteId=<excluded Id site>&text=<text>&start=<yyyy-MM-dd>&stop=<yyyy-MM-dd>&max=<max> &id=<id>&requestId=<requestId>**&@.displayName=<name>**

**Problem domain:**

CDS filters by status.

http://<host>:<port>/repositories.med.DNS /fpds//<templateId>/<filterId>/<patientId>/<domain>?clientName=<clientName>&excludedIdentifier=<excluded Id>&siteId=<excluded Id site>&text=<text>&start=<yyyy-MM-dd>&stop=<yyyy-MM-dd>&max=<max> &id=<id>&requestId=<requestId>**&@.statusName=<status>**

**Notes domain:**

CDS filters by facility name and/or Document Type name and/or local title and/or status.

http://<host>:<port>/repositories.med.DNS /fpds//<templateId>/<filterId>/<patientId>/<domain>?clientName=<clientName>&excludedIdentifier=<excluded Id>&siteId=<excluded Id site>&text=<text>&start=<yyyy-MM-dd>&stop=<yyyy-MM-dd>&max=<max> &id=<id>&requestId=<requestId>**&@.facilityName=<name>&@.documentTypeName=<name>&@.localtitle=<title>&@.statusName=<name>**

1. Approval Signatures

This section contains the reviews and approval signatures for the *HDR 3.15 VIP Build 1 System Design Document* *Volume 1.*

Peer Review:

AERB Waiver:

The signatures below indicate agreement and acceptance of the declarations contained in this document.

//es// Dennis Peterson

Dennis Peterson Date

Program Manager, Fix the Phones (FtP)

//es// Ellen Hans

Ellen Hans Date

Program Manager, Home Telehealth (HTH)

//es// Baron Woods

Baron Woods Date

Project Manager, HDR

Acting IT Program Manager, Repositories

1. Signature Verification

The Signature Verification section is used to verify and document the electronic signatures, concurrence and approval of the *HDR 3.15 VIP Build 1 System Design Document* *Volume 1.*