DATUP Update

System Design Document



Department of Veterans Affairs

**May 2015**

Version 6.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 05/27/2015 | 1.6 | Added ALT tags to graphics, moved Captions above the image. Updated Signature Block |  |
| 05/27/2015 | 1.5 | Added figure captions | , |
| 05/27/2015 | 1.4 | Updated TOC, fixed footer |  |
| 05/26/2015 | 1.3 | Tech edited |  |
| 05/22/2015 | 1.2 | Updated content |  |
| 04/13/2015 | 1.1 | Reviewed for Template Compliance |  |
| 01/31/2013 | 1.0 | Published and Signed, except for AERB |  |
| 01/28/2013 | 0.4 | Technical Writer Edits |  |
| 01/25/2013 | 0.3 | Technical Update to Various Sections. |  |
| 01/2013 | 0.2 | Document updated for DATUP 2.0. |  |
| 12/12/2012 | 0.1 | Document Created |  |

Artifact Rationale

The System Design Document (SDD) is a dual-use document that provides the conceptual design as well as the as-built design. This document will be updated as the product is built, to reflect the as-built product. Per the Project Management Accountability System (PMAS) Guide, the SDD as a conceptual design is required prior to the Milestone 1 Review. (Sections 1, 2, 3, 4, 5, 7, 9 need to be populated, as applicable.) The as-built design for each delivery must be incorporated prior to the Milestone 2 Review. (The entire document needs to be populated or updated, as applicable.)

Table of Contents

[1. Introduction 1](#_Toc420502664)

[1.1. Purpose of the SDD 1](#_Toc420502665)

[1.2. Identification 1](#_Toc420502666)

[1.3. Scope 1](#_Toc420502667)

[1.4. Constraining Policies, Directives and Procedures 1](#_Toc420502668)

[1.4.1. FDB DIF Database Technical Specification 2](#_Toc420502669)

[1.4.2. ONE-VA TRM Approved Methodologies and Tools 2](#_Toc420502670)

[1.5. User Characteristics 2](#_Toc420502671)

[1.6. Relationship to Other Documents and Plans 2](#_Toc420502672)

[1.7. Definitions, Acronyms, and Abbreviations 3](#_Toc420502673)

[2. Background 5](#_Toc420502674)

[2.1. Overview of the System 5](#_Toc420502675)

[2.2. Overview of the Business Process 5](#_Toc420502676)

[2.3. Business Benefits 5](#_Toc420502677)

[2.4. Assumptions and Constraints 5](#_Toc420502678)

[2.4.1. Design Assumptions 5](#_Toc420502679)

[2.4.2. Design Constraints 6](#_Toc420502680)

[2.4.3. Architectural Constraints 6](#_Toc420502681)

[2.4.4. Design Trade-offs 6](#_Toc420502682)

[2.5. Overview of the Significant Requirements 6](#_Toc420502683)

[2.5.1. Overview of Significant Functional Requirements 7](#_Toc420502684)

[2.5.2. Overview of Functional Workload / Performance Requirements 7](#_Toc420502685)

[2.5.3. Overview of Operational Requirements 8](#_Toc420502686)

[2.5.4. Overview of the Technical Requirements 8](#_Toc420502687)

[2.5.5. Overview of the Security or Privacy Requirements 8](#_Toc420502688)

[2.5.6. Overview of System Criticality and High Availability Requirements 8](#_Toc420502689)

[2.5.7. Single Sign-on Requirement 8](#_Toc420502690)

[2.5.8. Requirement for Use of Enterprise Portals 9](#_Toc420502691)

[2.5.9. Special Device Requirements 9](#_Toc420502692)

[2.6. Legacy System Retirement 9](#_Toc420502693)

[3. Conceptual Design 10](#_Toc420502694)

[3.1. Conceptual Application Design 10](#_Toc420502695)

[3.1.1. Application Context 10](#_Toc420502696)

[3.1.2. High-Level Application Design 12](#_Toc420502697)

[3.1.3. Application Locations 14](#_Toc420502698)

[3.2. Conceptual Data Design 14](#_Toc420502699)

[3.2.1. Project Conceptual Data Model 14](#_Toc420502700)

[3.2.2. Database Information 16](#_Toc420502701)

[3.3. Conceptual Infrastructure Design 16](#_Toc420502702)

[3.3.1. System Criticality and High Availability 17](#_Toc420502703)

[3.3.2. Special Technology 17](#_Toc420502704)

[3.3.3. Technology Locations 17](#_Toc420502705)

[3.3.4. Conceptual Infrastructure Diagram 19](#_Toc420502706)

[4. System Architecture 21](#_Toc420502707)

[4.1. Hardware Architecture 23](#_Toc420502708)

[4.2. Software Architecture 23](#_Toc420502709)

[4.3. Network Architecture 26](#_Toc420502710)

[4.4. Service Oriented Architecture / ESS 26](#_Toc420502711)

[4.5. Enterprise Architecture 27](#_Toc420502712)

[5. Data Design 29](#_Toc420502713)

[5.1. DBMS Files 29](#_Toc420502714)

[5.2. Non-DBMS Files 30](#_Toc420502715)

[5.3. Data View 30](#_Toc420502716)

[6. Detailed Design 31](#_Toc420502717)

[6.1. Hardware Detailed Design 31](#_Toc420502718)

[6.2. Software Detailed Design 31](#_Toc420502719)

[6.2.1. Conceptual Design 31](#_Toc420502720)

[6.2.2. Functional Area: Process Update File - National 31](#_Toc420502721)

[6.2.3. Functional Area: Process Update File - Local 33](#_Toc420502722)

[6.2.4. PECS and DATUP Coordination for FDB Incremental process 35](#_Toc420502723)

[6.3. Network Detailed Design 37](#_Toc420502724)

[6.4. Service Oriented Architecture / ESS Detailed Design 37](#_Toc420502725)

[7. External System Interface Design 38](#_Toc420502726)

[7.1. Interface Architecture 38](#_Toc420502727)

[7.2. Interface Detailed Design 40](#_Toc420502728)

[8. Human-Machine Interface 41](#_Toc420502729)

[9. Security and Privacy 42](#_Toc420502730)

[9.1. Security 42](#_Toc420502731)

[9.2. Privacy 42](#_Toc420502732)

[A.1. RTM 44](#_Toc420502733)

[A.2. Packaging and Installation 46](#_Toc420502734)

[A.3. Design Metrics 46](#_Toc420502735)

[A.4. Required Technical Documents 46](#_Toc420502736)

# Introduction

DATUP (Data Update) is a utility that runs an automated process to maintain the FDB-DIF and VA custom data used by regional MOCHA instances and at the National level by PECS and PPS-N.

## Purpose of the SDD

The purpose of this document is to describe in sufficient detail how the DATUP Utility is to be constructed. This document translates the Requirements Specifications into a document from which the developers can create the DATUP Utility. It identifies the top-level system architecture, as well as hardware, software, communication, and interface components.

The intended audience of this System Design Document (SDD) is the Pharmacy Reengineering (PRE) project team, SQA, and other teams or departments that may be interested parties to the development of this utility.

## Identification

This document describes the DATUP utility.

## Scope

The scope of this document includes the functionality applicable to the DATUP Utility, including design and implementation details on the significant or important logical parts of the application. The run-time use is for updating the data in the National and Local Pharmacy Enterprise Product System databases as well as reporting on the status of those updates.

Table 1: Scope Inclusions

|  |
| --- |
| Includes |
| Process Update File – National |
| Process Update File – Local |
| Retrieve Update Files from FDB |

Table 2: Scope Exclusion

| Excludes |
| --- |
| Generating and making available Custom Update files from PECS |

## Constraining Policies, Directives and Procedures

Various VA policies, directives, and procedures are used as guidelines/recommendations for the PECS Application design.

***Note:*** *Due to policy constraints, active links cannot be included in this document. Please copy and paste the URLs into your browser.*

* Section 508 Checklist for Web-based Internet Information and Applications  
  http://vista.domain.ext/508workgroup/docs/web\_508\_checklist\_ver\_03\_10\_10.doc
* One-ONE-VA TRM Approved Technologies/Standards List  
  http://trm.oit.domain/ApprovedToolListSummaryPage.asp
* Java Coding Standards v2.0  
  http://vista.domain.ext/vistaarch/healthevet/Documents/Java%20Coding%20Standards.doc

Various VA policies, directives, and procedures specify constraints that the DATUP utility design must conform to. There are two basic constraints, detailed in the sections below:

* The FDB DIF database technical specification.
* ONE-VA TRM (Technical Reference Model) approved methodologies and tools

### FDB DIF Database Technical Specification

DATUP uses a Commercial Off-the-Shelf (COTS) Drug Database from FDB. This product is designed for use for medication checks through an API (MOCHA Server) and the technical specification for the database schema is not available for reference. Aside from the DIF tables, other database objects such as foreign keys, indexes, sequences, etc., are not known. DATUP will not access the database directly but relies on the FDB Data Updater classes to update the data in the database.

### ONE-VA TRM Approved Methodologies and Tools

Specific tools and methodologies used in the implementation of the proposed solution must adhere to the VA Technical Reference Model (One-ONE-VA TRM). Specifically, any technologies used must be on the TRM Approved Product List.

## User Characteristics

The system’s user community is primarily comprised of the Pharmacy Benefits Management (PBM) group, ADPAC users, and a National Drug File (NDF) manager or designee and administrators and operators at the site.

## Relationship to Other Documents and Plans

Note: Due to policy constraints, active links cannot be included in this document. Please copy and paste the URLs into your browser.

This SDD relies on the following documents:

* DATUP Requirements Specification Document (RSD)

http://vaww.yourserver.domain/projects/pre/PRE\_IPT\_Rev/PRE\_IPT\_Rev\_DATUP/SitePages/Home.aspx

* Pharmacy Reengineering (PRE) Configuration Management Plan (CMP)

http://vaww.yourserver.domain/projects/pre/OverArching%20Documents/PRE\_Config\_Mgmnt\_Plan.doc

* PECS SDD

http://vaww.yourserver.domain/projects/pre/PRE\_IPT\_Rev/PRE\_IPT\_Rev\_PECS/Lists/Links/AllItems.aspx

• MOCHA Server SDD

http://vaww.yourserver.domain/projects/pre/PRE\_IPT\_Rev/MS1-1\_Review/default.aspx

• PECS Interface Control Document

http://vaww.yourserver.domain/projects/pre/PRE\_IPT\_Rev/PRE\_IPT\_Rev\_PECS/Lists/Links/AllItems.aspx

• MOCHA Server Interface Control Document

http://vaww.yourserver.domain/projects/pre/PRE\_IPT\_Rev/MS1-1\_Review/default.aspx

## Definitions, Acronyms, and Abbreviations

The following table defines terms and acronyms used throughout the document.

Table 3: Definitions, Acronyms and Abbreviations

| Term | Definition |
| --- | --- |
| ADPAC | Automated Data Processing Application Coordinator |
| AITC | Austin Information Technology Center |
| API | Application Program Interface |
| CDCO | Corporate Data Center Operations |
| COTS | Commercial Off-the-Shelf |
| COOP | Continuity of Operations |
| CRUD | Create, Read, Update, Delete functions |
| DATUP | Application that implements the FDB-DIF update business logic using the FDB Updater APIs to process the update file |
| DAO | Data Access Object |
| DBA | Database Administrator |
| DBMS | Database Management System |
| DDL | Data Definition Language. A computer language for describing the records, fields, and "sets" making up a database. |
| DR | Disaster Recovery |
| EA | Enterprise Architecture |
| EAR | J2EE Enterprise Application Archive file. |
| FDB DIF | First Databank Drug Information Framework database |
| FOD | Field Operations and Development |
| SFTP | Secure File Transfer Protocol |
| GCNSeqNo | Generic Code Number Sequence Number |
| GUI | Graphical User Interface |
| HA | High Availability |
| ISO | Information Security Officer |
| JEE | Java Enterprise Edition |
| HWSC | Healt*h*eVet Web Services Client |
| MVC | Model-View-Controller |
| NDF | National Drug File |
| OIFO | Office of Information Field Office |
| PBM | Pharmacy Benefits Management |
| PECS | Pharmacy Enterprise Customization System |
| PITC | Philadelphia Information Technology Center |
| PRE | Pharmacy Reengineering |
| RDBS | Relational Database System |
| RSD | Requirements Specification Document |
| SDE | System Design & Engineering |
| SQA | Software Quality Assurance |
| SQL | Structured Query Language |
| TRM | Technical Reference Model |
| UI | User Interface |
| URL | Uniform Resource Locator |
| VA | Department of Veterans Affairs |
| VAMC | Veterans Affairs Medical Center |
| VistA | Veterans Health Information Systems and Technology Architecture |

# Background

To meet the current and future business needs of the Department of Veterans Affairs (VA) Pharmacy, and to support the overall architecture planned for HealtheVet VistA, the existing pharmacy software modules are being re-engineered through new development and the purchase of Commercial Off-the-Shelf (COTS) products.

Transition to the new Pharmacy Reengineered system eliminates the VistA Pharmacy files currently referenced.

## Overview of the System

The DATUP Java Utility is designed as a Java Enterprise Edition Application (JEE), which synchronizes the data in PEPS databases.

## Overview of the Business Process

The business processes implemented within DATUP include the ability to synchronize the data included in the National and Regional databases in use by various PEPS applications. Local/Regional databases are being utilized simply for performance purposes. DATUP must be able to accept update from FDB and PECS and be able to process those files in order to apply the data to the respective databases.

The table below provides an overview of the business processes that DATUP will support.

Table 4: Business Practices

| Business Process ID | | Business Process Name | Owner – Organization Performing the Process | Description |
| --- | --- | --- | --- | --- |
| *1* | Process Update File - National | | The owners are the doctors and pharmacists at the regional or local site where the data is loaded | The process of updating a National PEPS HealtheVet database. |
| *2* | Process Update File – Local | | The owners are the doctors and pharmacists at the regional or local site where the data is loaded | The process of updating a Local/Regional PEPS HealtheVet database. |

## Business Benefits

DATUP allows for the automated updating of data updates initiated by FDB or by PECS. This keeps the data used in the MOCHA Server Order Check processing the same across all instances.

## Assumptions and Constraints

The following section describes the assumptions and constraints that impacted the design of the DATUP utility:

### Design Assumptions

The System design assumptions are that the application:

* Must be compliant with framework/tools/library as recommended by ONE-VA TRM.
* Must make use of VA Common services – VistA Link/HWSC.

### Design Constraints

All VA designated constraints for design will be followed.

### Architectural Constraints

In order to deploy and use DATUP to perform the new Pharmacy business processes, other systems being developed by the VA for the HealtheVet-VistA architecture must be in place and operational. The details of the systems with which PRE must interface are defined in the PECS and MOCHA Server Interface Control Documents (ICDs). The following subsections contain brief descriptions of the architecturally significant constraints identified to date.

#### Architectural Design

The DATUP architecture is based on the VHA HealtheVet architecture guidelines established by the VHA Health Information Architecture group as well as the software systems architectural requirements based on the concept of the ONE-VA Enterprise Architecture. These guidance groups are continually refining the requirements of the systems in the VA enterprise. PRE’s DATUP adheres to those known requirements published by these groups and the SDE group. Because these requirements are constantly being changed and refined, and because some groups publish requirements that other groups later challenge, DATUP can only adhere to those requirements published in sufficient time for the design and implementation to be conducted.

#### Deployment Environment

The VHA HealtheVet-VistA architecture’s definition of the physical hardware to be used for HealtheVet-VistA systems, including the network architecture for these new software systems. A draft DATUP deployment design was delivered to the SDE group and approved for use by DATUP. This approved, draft DATUP deployment design is the driving force for the DATUP deployment architecture.

#### Physical HealtheVet-VistA Architecture

PRE does not have the responsibility of defining the physical hardware infrastructure for HealtheVet-VistA. DATUP co-exists with other application domains within the VHA Enterprise Architecture. In order for PRE’s DATUP to deploy, the hardware and network architecture must be in place and accessible to PRE. The architecture spelled out in this document makes no assumptions about the physical architecture but does contain assumptions about the logical architecture.

### Design Trade-offs

N/A

## Overview of the Significant Requirements

This section is not intended to replace either the existing functional and technical requirements documents, nor to serve as the basis for the Requirements Traceability Matrix; its purpose is to inform non-project personnel reading this document of the basis for the Application design.

### Overview of Significant Functional Requirements

This section provides an overview of the major functional requirements for the system. The goal is not to include the full set of requirements in this document or to replace the functional requirements documents, but to identify the major functions to be performed and the few major requirements that drive the design described in the sections below. The emphasis is on identifying the impact that these requirements have on the design. The requirements below are synopsis of the major requirements.

Table 5: System Requirements

| ID | Requirement |
| --- | --- |
| SPEC1196 | FDB-DIF incremental files are processed by the date sequences, from the oldest to newest if more than one file is available for processing. |
| SPEC1197 | Order Check Validation is done before and after the automated data update to ensure integrity of FDB-DIF Drug database. |
| SPEC1198 | The application should automatically check daily for available FDB-DIF update files received from CMOP files to process. |
| SPEC1199 | The application should automatically check daily for available VA Custom Incremental files to process. |
| SPEC1200 | The application should be configurable, for each installation, to begin the update process at a configured time if the FDB and/or VA incremental updates are present. |
| SPEC1201 | Validation of Success or Failure of the Updates . |
| SPEC1202 | The application should provide automated email notifications of success or failure of the update. The email addresses group for these notifications is configurable. |
| SPEC1203 | The application should be able to manage the FDB-DIF and VA custom files stored nationally and at the region or local sites. |
| SPEC1204 | The application should be able to upload Images that are received with the weekly FDB-DIF updates to the PECS/PPSN Production Server (National). |

### Overview of Functional Workload / Performance Requirements

No specific workload or performance requirements have been documented for DATUP.

### Overview of Operational Requirements

There are no specific operational requirements that drive the system design. The DATUP project is in compliance with Enterprise System Engineering and has been submitted for the Systems Engineering and Design Review (SEDR) process (and received appropriate guidance from ESE/SDE).

### Overview of the Technical Requirements

The table below displays the major technical requirements from the ONE-VA TRM that drive the DATUP conceptual design.

Table 6: Technical Requirements

| ID | Requirement |
| --- | --- |
| 1 | WebLogic will be used as the application server. |
| 2 | Oracle Database will be used as the RDBS (Relational Data Management System) for DATUP at the National Level. |
| 3 | JEE Framework will be used for the application development. |

The DATUP Utility is part of the larger Pharmacy Reengineering (PRE) domain and is a continuation of an ongoing VA project started in 2008 (PRE v 0.5).

At the time of release, the DATUP technologies are in compliance with the ONE-VA TRM.

### Overview of the Security or Privacy Requirements

PD PRE will user12 with its Security Engineer and the SDE CDCO designated ISO on the need to pursue recertification and accreditation. PRE’s position is that a change in the underlying database alone should not require this, but we will ask for an official opinion and take the appropriate action based on the direction given.

### Overview of System Criticality and High Availability Requirements

The system shall be available 24/7, with the exception of required system maintenance activities. Required maintenance activities shall be scheduled for known periods of decreased system utilization.

SDE CDCO in conjunction with FOD (and PRE) will address HA COOP / DR for DATUP for the Regional Server Clusters hosted at AITC and PITC (managed by Regional Field Operations).

### Single Sign-on Requirement

N/A

### Requirement for Use of Enterprise Portals

N/A

### Special Device Requirements

There are no special devices for the DATUP utility.

## Legacy System Retirement

The table below identifies each of the mainframe application components that will be retired or will have their workloads significantly reduced as a result of the design and deployment of this system.

Table 7: Proposed Legacy Retirements

| Legacy System or  Legacy System Component | System Retired or  Workload Reduced | Quantify the Workload Reduction |
| --- | --- | --- |
| DRUG INTERACTION file (#56) | This file is used for Drug Interaction Order Checks, it is being updated by PBM through NDF updates. There are a couple of supported APIs for this file which may be used by other applications. |  |
| Enter/Edit Local Drug Interaction option (Pharmacy Data Management application (PSS\*) | Deleted |  |
| Edit Drug Interaction Severity option (PSS\*) | Deleted |  |

# Conceptual Design

## Conceptual Application Design

### Application Context

The diagram below is showing the context within which the application exists*.* The following figure depicts the DATUP utility and the external system(s) that it interacts with**.**

Figure 1: DATUP Utility Context Diagram



The table below describes the drawing above as follows: it describes each of the objects in the drawing and then displays the context of each of the interfaces, external and internal, to the Office of Information & Technology, which must be aware of all VA applications.

The table below describes the information in the Application Context Diagram in four sections. Note that since the system for which this design applies is represented by a single object (typically in the center of the diagram), it is not referred to in the table below.

Table 8: Application Context Description

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Objects | | | | |
| ID from Drawing | Name | Description | | Interface Name | Interface System |
|  | MOCHA Server | Used to generate and execute Order Checks to validate the data was applied correctly. | | RandomOrderCheckCapability  ProcessOrderChecksCapability | *Service* |
|  | National FDB DIF Data Store | Store Drug information for enhanced Drug Order Checks at the National level. | | JDBC | *Data Store* |
|  | Local FDB DIF Data Store | Store Drug information for enhanced Drug Order Checks at the Local/Regional level. | | JDBC | *Data Store* |
|  | Interfaces External to OI&T | | | | |
| ID | Interface Name | Related Object | Input Messages | Output Messages | External Party |
|  | N/A |  |  |  |  |
|  | Interfaces Internal to OI&T | | | | |
| ID | Interface Name | Related Object | Input Messages | Output Messages | Other CBP Party |
|  | National Manager | e-mail  Reports | Request for Report | Report  Update status e-mail | N/A |
|  | Local/Regional Manager | e-mail | N/A | Update status e-mail | N/A |
|  | Externally Shared Data Stores | | | | |
| ID | Name | Data Stored | Owner | Access |  |
|  | SFTP Server | File System | PBM | Read/Write | N/A |

### High-Level Application Design

The High Level Application Design identifies the major components of the application or utility and the relationships of its major components to each other and to the surrounding applications. The major components of the utility are at the “subsystem” or top level service area and are depicted in the diagram.

The figure below provides a high level dataflow diagram for the PECS system.

Figure 2: Sample High-Level Application Design



Table 9: Objects in the High Level Application Design

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Objects** | | | | | | | | |
| **Name** | **ID** | **Description** | **Service or Legacy Code** | **External Interface Name** | **External Interface ID** | **Internal Interface Name** | **Internal Interface ID** | **SDP Sections 1&2** |
| Enhanced Drug Order Check Query | *1* | The *Query Service* is used to Query Drug Information from FDB-DIF Database. | Service | N/A | N/A | The internal call to the *Query Service* is performed via Java method calls |  |  |
| FDB\_DIF-VA Custom Updates | *4* | *DATUP* is used to update Incremental FDB-DIF Drug-Data and VA Custom Drug data. | Interface | N/A | N/A | Same as above |  |  |
| DATUP | *5* | *DATUP* is used to update Incremental FDB-DIF Drug-Data and VA Custom Drug data for five drug concepts. *DATUP* pulls the updates from the National SFTP Server. | Interface (application system) | N/A |  | The internal call performed via Java method calls |  |  |
| Legacy MOCHA | *3* | The Legacy MOCHA Application uses MOCHA Server to Query Drug Information from FDB-DIF Database | Legacy | N/A | N/A | N/A |  |  |
|  |  |  |  |  |  |  |  |  |

Table 10: Internal Data Stores

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | ID | Data Stored | Steward | Access |
| FDB-DIF Database | 2 | The MOCHA Server FDB-DIF database is a COTS database which contains the drug-data information. | FDB-DIF | The MOCHA Server database is a COTS database which contains the drug-data information |

### Application Locations

The following table denotes the location of the components in the DATUP Utility.

Table 11: Application Locations

|  |  |  |  |
| --- | --- | --- | --- |
| Application Component | Description | Location at Which Component is Run | Type |
| DATUP National | *DATUP* is used to update Incremental FDB-DIF Drug-Data and VA Custom Drug data for five drug concepts. DATUP pulls the updates from National SFTP Server. | Information Technology Center (ITC) – DATUP Application Server | Business Logic |
| DATUP Local | *DATUP* is used to update Incremental FDB-DIF Drug-Data and VA Custom Drug data for five drug concepts. DATUP pulls the updates from National SFTP Server. | Regional Sites | Business Logic |
| FDB-DIF Database – National | Stores Drug information for the PECS application | Information Technology Center (ITC) – PECS Database Server | Data Logic (via Oracle) |
| FDB-DIF Database – Local/Regional | Stores Drug information for MOCHA Server. | Regional Sites | Data Logic (via Caché) |

Table 12: Application Users

|  |  |  |
| --- | --- | --- |
| Application Component | Location | User |
| DATUP National | Austin Information Technology Center (ITC) – PECS Database Server | Pharmacy Benefits Management (PBM) group, ADPAC users, NDF manager or designee. |
| DATUP Local | Local/Regional Sites | FOD |

## Conceptual Data Design

### Project Conceptual Data Model

DATUP utilizes First Databank’s (FDB) Drug Information Framework (DIF) database. FDB\_DIF is a COTS product provided by FDB and DATUP uses FDB Data Updater APIs to access the FDB\_DIF Drug database.

Note: Due to policy constraints, active links cannot be included in this document. Please copy and paste the URLs into your browser.

A high level model for the FDB\_DIF database is provided in the figure below: (This for reference only, please refer to FDB documentation for details:- (IMPLEMENTATION GUIDE - DRUG

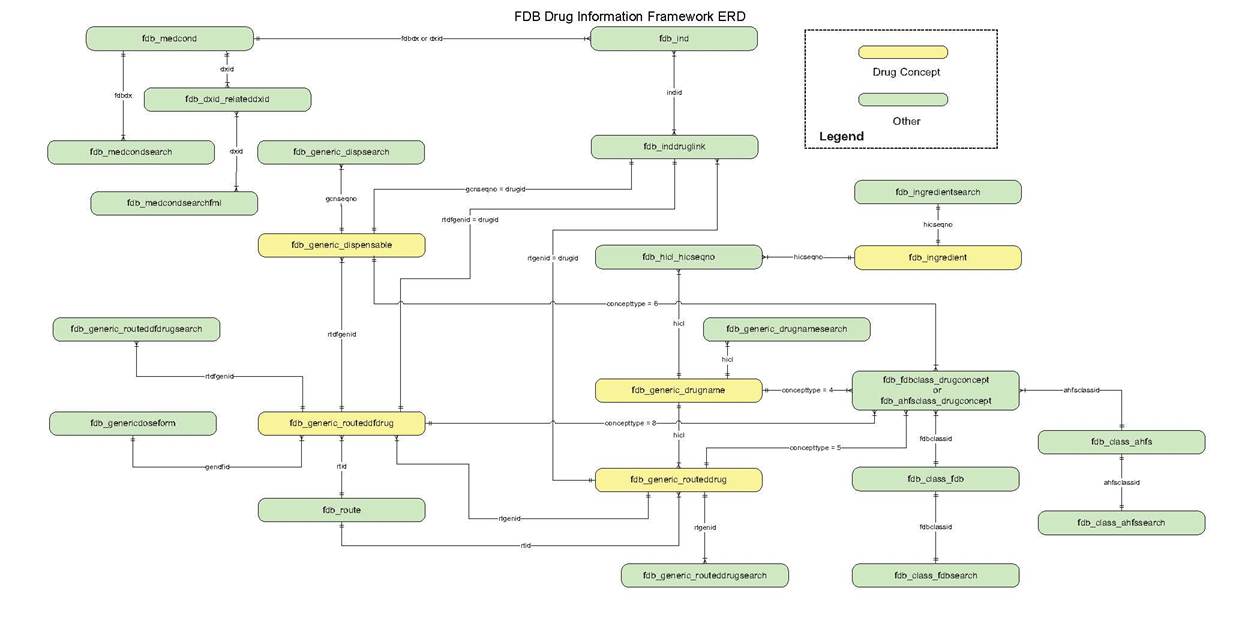
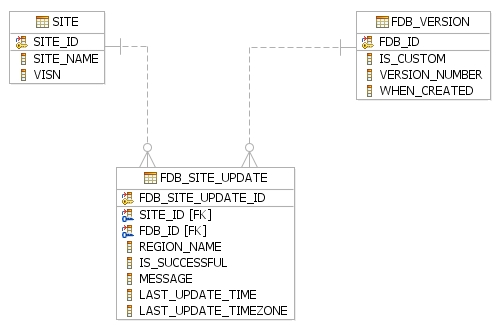


Figure 3: Sample Project Conceptual Datal Model

The tables that DATUP will use for reporting purposes are shown in the following diagram.

Figure 4: FDB-DIF Conceptual Data Model



### Database Information

The table below provides a listing of all databases that DATUP interacts with.

Table 13: Database Inventory

|  |  |  |  |
| --- | --- | --- | --- |
| Database Name | Description | Type | Steward |
| FDB\_DIF | First Databank Drug Information Framework database | Interface | PRE |

## Conceptual Infrastructure Design

DATUP is a Java utility- implemented as JEE. There are no interfaces to DATUP.

The figure below shows the high level overview of the logical deployment design for DATUP.

**Application Server:**

There will be a J2EE Application Server - WebLogic Application Server 10.3. The Application Server will have Red Hat Linux Enterprise version RHEL5 as its operating system. The WebLogic server will host POCS and its business services.

**Data Base Server** :

There will be an InterSystems Caché Database Server. The Database Server will have Red Hat Linux Enterprise version RHEL5 – as its operating system and Oracle version Oracle 11.2.0.3. It will host the FDB\_DIF database.

**Failover Server**:

There will be a Failover server. The Database Server will have Red Hat Linux Enterprise version RHEL5 – as its operating system.

**Legacy Interface**:

There will be an existing VistA server which will host the legacy Pharmacy system API.

Figure 5: Logical Deployment Design of DATUP



### System Criticality and High Availability

DATUP, an essential utility for the Pharmacy Benefits program, high availability is required. For Disaster Recovery Plans, please refer to the Disaster Recovery (DR) document that is developed by SDE/FOD and Continuity of Operations (COOP) Design Document Plan for additional information.

### Special Technology

No special technology is employed for the DATUP utility.

### Technology Locations

The table below describes the various technology components used in the DATUP utility. The MOCHA Server Production environment will be hosted at the ITC, Austin, TX. For VA SQA testing, SQA servers will be located at the Bay Pines, FL, and AITC.

Table 16: Technology Locations and Details – Production 1

| Technology Component | Location | Usage |
| --- | --- | --- |
| Workstations | VAMC | MOCHA Legacy Application users will log into the MOCHA application via CPRS application on these workstations |
| Special Hardware | N/A |  |
| Interface Processors | N/A |  |
| Legacy Mainframe | N/A |  |
| Legacy Application Server | VAMC | Legacy MOCHA application is hosted on VistA Servers at VAMC’s |
| Legacy Databases | VAMC | Legacy VistA database is hosted on VistA Server |
| HeV Application Server | AITC/PITC/Local/Regional | WebLogic Application Server – hosts DATUPutility |
| HeV Database Server | AITC/PITC | Oracle Database Server – hosts backend FDB\_DIF database at National |
| HeV Database Server | Local/Regional | Caché Database Server – hosts backend FDB\_DIF database at Local/Regional sites |

### Conceptual Infrastructure Diagram

The following two sub-sections describe and depict the DATUP conceptual networks and environments, and DATUP’s conceptual production configuration.

#### Location of Environments and External Interfaces

The figure below shows the Production environment that will be supported, and the local networks to which they will be attached for Local VAMCs, where MOCHA users are located.

Figure 6: MOCHA Server Conceptual Networks and Environments



Conceptual Production String Diagram

The figure below shows high level conceptual production configuration expected for the MOCHA Server application. The MOCHA Server application is deployed on a WebLogic server, and handles all Drug Order check requests from Legacy MOCHA via HWSC.

Figure 7: MOCHA Server Production String Diagram



# System Architecture

DATUP is a Java-based (JEE) application that is part of the VHA’s HealtheVet infrastructure.

The DATUP Logical architecture includes the provision for WebLogic application server which hosts DATUP JEE application, the FDB-DIF database will be hosted on Oracle Database Server Cluster.

In the figure, VistA resides on the left side of the vertical line while HealtheVet resides on the right side of the vertical line. As the figure shows, communication between DATUP and VistA is via the PRE Encapsulation layer, including all requests for MOCHA data from all entities within VistA (a.k.a. Systems Engineering and Design Review of PEPS).

Figure 8: Overview of Logical Architecture Design



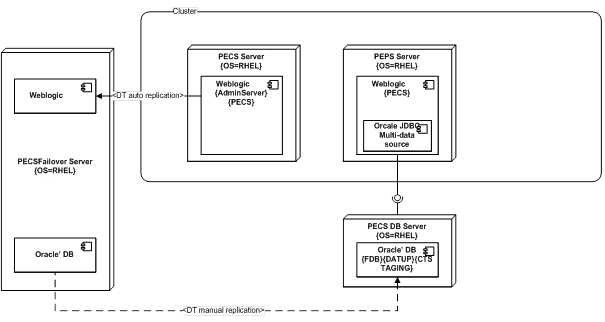
Figure 9: System Context of PRE MOCHA Server



## Hardware Architecture

The section below describes the system hardware architecture and the hardware modules. The diagram shows the connectivity between the modules.

Figure 10: Deployment Solution by SDE (ESE)



The figure above depicts the Deployment Solution proposed by VA Enterprise System Engineering (ESE) for PEPS (MOCHA Server) in December 2010.

This design uses a single WebLogic cluster for the PRE application, and a multi-data source pool for all Oracle data sources. The Failover server will only replicate the WebLogic server with the AdminServer. Upon failover the FO WebLogic instance would then assume the AdminServer/Application Server functions. The Failover Server will only need to replicate a single Oracle data source.

The PRE application is stateless and the FDB database is read-only. Updates to the database will occur in off hours manually and using a process that would allow the other data sources to maintain the availability of the PRE service. WebLogic Multi-data source would allow for the Oracle data source to be shared among all the Cluster members, thus providing the HA capability without the need for an additional Cluster for the data source servers.

## Software Architecture

The section below describes the overall system software and organization. It lists and describes the software modules, programming languages, and development tools.

DATUP is a Java Utility – implemented as a JEE application that is part of the VHA’s HealtheVet infrastructure. The DATUP implementation is based on n-tier architecture; ), the application is mid-tier, which is a Java Enterprise Application deployed on a J2EE server, and the data-tier composed of a backend Oracle or Caché database.

The DATUP Logical architecture includes the provision for a Local PRE system. In the proposed architecture, the FDB-DIF database will be hosted on Caché Database Server at the Regional level. DATUP will involve updating the Local instance of FDB-DIF database with data from the COTS (FDB-DIF) drug data.

In the figure below, VistA resides on the left side of the vertical line while HealtheVet resides on the right side of the vertical line. As the figure shows, communication between DATUP (PEPS system) and VistA is via the PRE Encapsulation layer, including all requests for MOCHA (a.k.a. PEPS) data from all entities within VistA.

When a VistA package requests an order check, the data required to perform the check will be sent from VistA to FDB via VistALink in the form of an extensible mark-up language (XML) message. The PRE MOCHA services will receive and validate the format of the request. Provided the format is correct, the PRE MOCHA services will triage the request, interacting with FDB to perform the requested check, and return the results.

An overview of the approach is shown in the figure below. The components that reside in the current VistA system are shown on the left while the HealtheVet components are shown on the right. Communication between the two components is via VistALink/HWSC.

On the VistA side, the VistA Drug Check API will support drug-drug, drug-therapy, and duplicate drug checks in addition to maximum single dose and daily dose minimum and maximum checks.

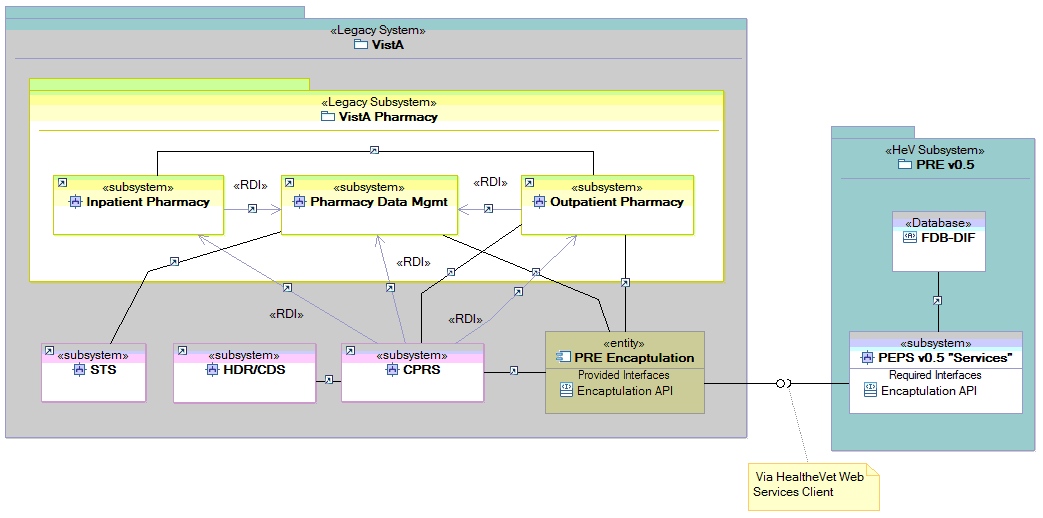
On the DATUP side (the green components on the right side of the figure), all three service components would be implemented.

The figure below depicts the overall Software Architecture for PRE DATUP, including MOCHA and VistA.

Figure 11: Overview of the DATUP & MOCHA Server Software Architecture Design



Figure 12: MOCHA Logical Deployment



When a VistApackage requests an order check, the data required to perform the check will be sent from VistAto FDB via VistALink in the form of an extensible mark-up language (XML) message. The DATUP services will receive and validate the format of the request. Provided the format is correct, the DATUP services will triage the request, interacting with FDB to perform the requested check, and return the results. (Note that in the figure above on the right side, the labels for “PRE v.05” and “PEPS v0.5 ‘Services’” should be replaced with “PECS” and “DATUP,” respectively.)

The link between a VistAdrug concept and an FDB drug concept will be primarily the Generic Code Number Sequence Number (GCNSEQNO). If the GCNSEQNO cannot be mapped, if the drug is not matched to NDF or if cannot find a GCNSEQNO, no enhanced order checking (drug-drug, duplicate therapy or dosing checks) will be performed for the drug. An error message will be displayed to the user at that point for their action (to match to NDF) or to direct to National. However, the remaining drugs that are successfully mapped to an FDB drug concept will be checked. The return message will include the results of the check that was performed, as well as a list of the drugs that could not be mapped and were not checked.

## Network Architecture

DATUP communications architecture will use a combination of wide area networks (WAN) – VA Intranet coupled with local area networks (LAN) as depicted in the figure below. The LANs will use a star topology with transportation communication protocol and internet protocol (TCP/IP). The LAN will use gigabit or greater switches where required. All nodes within the confines of the ITC- Austin TX will be connected to the ITC- Austin TX LAN. All nodes within the confines of the Local VAMC facility will be connected to the Local LAN. A firewall is used to provide security and connect each LAN to the VA Intranet. Access to the MOCHA Legacy application is accomplished via a standalone laptop or workstation on the VA Intranet.

Figure 13: PECS Communication Architecture

**

## Service Oriented Architecture / ESS

N/A

## Enterprise Architecture

The adherence to technical standards is established means to provide for a consistent enterprise architecture. DATUP 3.0 is TRM compliant. TRM compliance is a major feature of the development plan and corrects the current deviancies from the TRM, the most significant of which is the replacement of FTP with SFTP. Below software that was upgraded between DATUP 2.0 and 3.0 to remain TRM compliant.

|  |  |  |
| --- | --- | --- |
| **Technology** | **Description** | **Version** |
| commons-collections | The Commons is an Apache project focused on all aspects of reusable Java components. | 3.2.1 |
| commons-configuration | 1.10 |
| commons-lang | 2.6 |
| commons-lang3 | 3.3.2 |
| commons-logging | 1.2 |
| commons-net | 2.0 |
| commons-vfs2 | 2.0 |
| DIF | First Databank MedKnowledge Java API | 3.3 |
| FDBConnection | 5/24/2010 - Last build date - no version identifier |
| FDBDataUpdater | 1.1 |
| jaxb-api | Java Architecture for Extensible Markup Language (JAXB RI) provides a reference implementation for Extensible Markup Language (XML) Binding JAXB specification | 2.2.7 |
| jaxb-impl | 2.2.7 |
| jsr173 | The Streaming API for XML (StAX) is a Java based API for pull-parsing XML | 1.0 |
| ps\_ms\_cfg | MOCHA Server libraries | 1.1.00.001 |
| ps\_ms\_common | 1.1.00.001 |
| ps\_ms\_common\_impl | 1.1.00.001 |
| spring-aop | Spring is an application framework based on Inversion of Control. It provides a lightweight Java Enterprise Edition (Java EE) container. Spring is a modular framework for Java applications that can be used in a stand-alone Java Virtual Machine (JVM) with the Java Standard Edition libraries, in servlet containers, and in Java EE servers. | 4.1.5 |
| spring-beans | 4.1.5 |
| spring-context | 4.1.5 |
| spring-context-support | 4.1.5 |
| spring-core | 4.1.5 |
| spring-expression | 4.1.5 |
| spring-jdbc | 4.1.5 |
| spring-tx | 4.1.5 |
| spring-web | 4.1.5 |

# Data Design

The DATUP application uses a relational back-end Oracle or Caché database. The FDB DIF database-COTS database is used to retrieve reference data and lookup values on which to base Order Check customizations. Since the FDB DIF database is a COTS application and is well documented, it will not be discussed here.

## DBMS Files

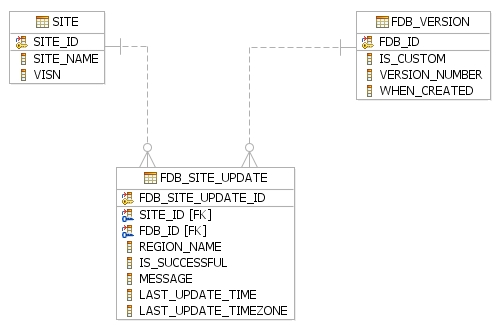
DATUP Application utilizes First Databank’s (FDB) Drug Information Framework (DIF) database. FDB\_DIF is a COTS product provided by FDB and DATUP uses FDB API’s to access the FDB\_DIF Drug database.

A high level model for the FDB\_DIF database is provided in the figure below: (This for reference only, please refer to FDB documentation for details.

Since the FDB DIF database is a COTS application and is well documented, it will not be discussed in detail here.

DATUP tables for reporting purposes are as follows:

Figure 14: DATUP Tables for Reporting, #2



FDB\_SITEUPDATE and Site table could be removed

## Non-DBMS Files

DATUP utilizes update files from FDB or PECS. These files will reside on the SFTP server that both DATUP National and DATUP Local have access to. Since these files are in a proprietary format defined by FDB, please refer to the FDB documentation for the specifics of the file formats.

## Data View

N/A

# Detailed Design

The sections below describe the proposed design in detail. They provide the necessary information for the development team to integrate the hardware components and write the software code, so that the hardware and software components will provide a functional DATUP product.

## Hardware Detailed Design

Refer to Section 4.1.

## Software Detailed Design

The section below provides enough detailed information for the developers to write the source code for all modules in the DATUP system and to integrate FDB-DIF COTS software product.

### Conceptual Design

The section below provides enough detailed information for the developers to write the source code for all modules in the DATUP system and to integrate FDB-DIF COTS software product.

### Functional Area: Process Update File - National

The Process Update File - National functional area allows for the updating of a National FDB DIF data store with current data from FDB or custom data from PECS. It also moves Full Custom and Full FDB files from the FDB folder to a “Full” folder.

#### Domain Classes

There are no domain classes for the Process Update File – National functionality

#### Data Layer

No custom classes are needed for DATUP in order to access the data. Only objects provided by the COTS FDB DIF API will be uses to access the data.

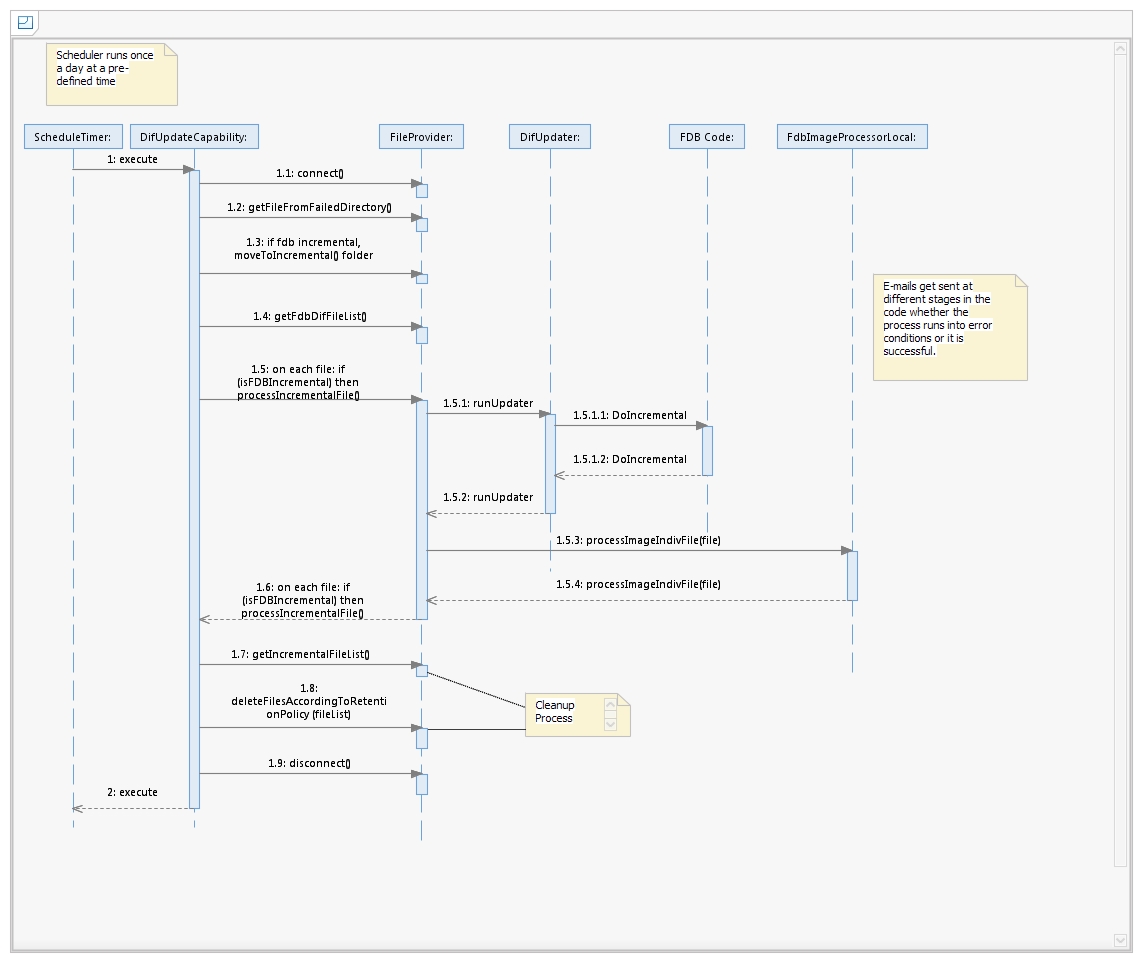
Business Layer

The business logic for the Process Update File – National functionality is to determine if a file containing update data should be applied to the database or not. It also performs the move of full files out of the folder where cmops stores them. The processing of files on DATUP National is performed in two steps. On the first step, it moves the full files and performs the database update on the VA Custom Incremental file. On the second step it performs the database update on the FDB Incremental file. The reason for breaking the process in two steps is that the FDB Incremental file processing depends on whether PECS had actually processed the file before DATUP makes use of it and removes it from the folder. Please refer to section 6.2.4 for an explanation on how this coordination between PECS and DATUP happens. Meanwhile the following sequence diagrams document the logic in each of these steps.

Figure 15: Moving Full files and Custom Incremental database update process

TUP Tables for Reporting, #2
 


Figure 16: FDB Incremental database update process



Presentation Layer

There are no presentation layer classes for the Process Update File – National functionality.

### Functional Area: Process Update File - Local

The Process Update File - Local functional area allows for the updating of a Local/Regional FDB DIF data store with current data from FDB or custom data from PECS.

Domain Classes

There are no domain classes for the Process Update File – Local functionality

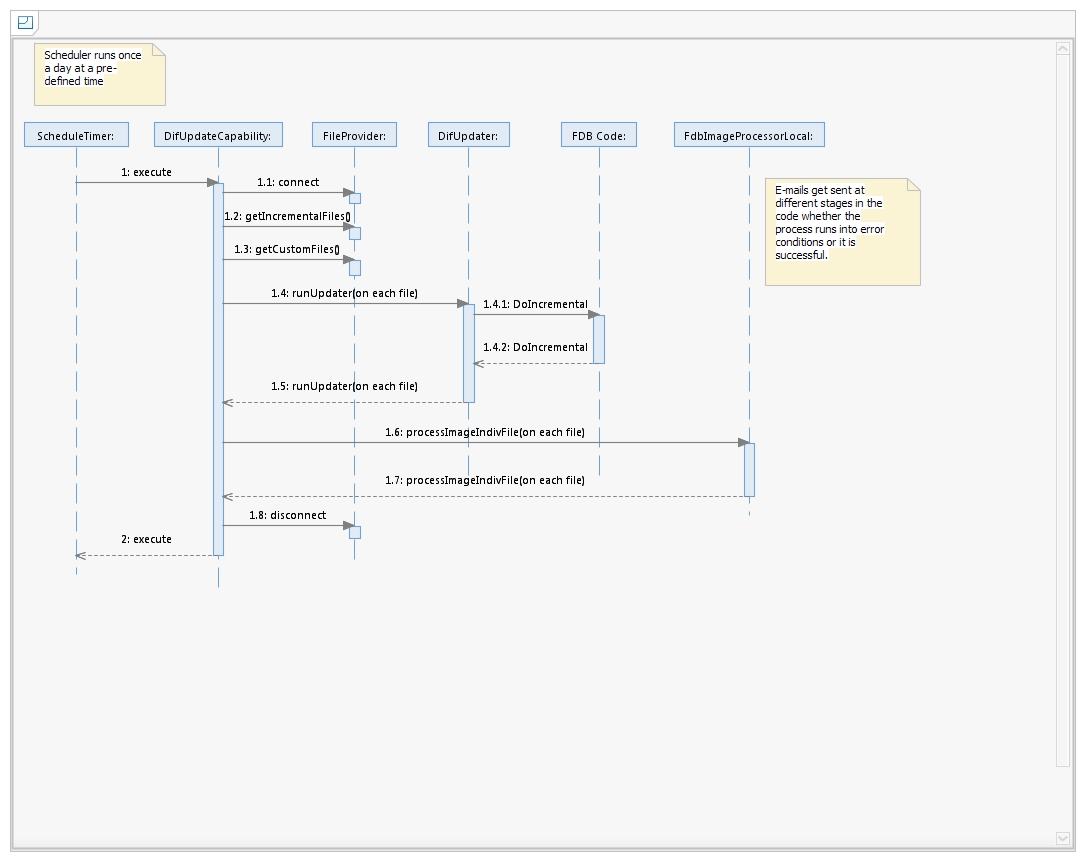
Data Layer

No custom classes are needed for DATUP in order to access the data. Only objects provided by the COTS FDB DIF API will be used to access the data.

Business Layer

The business logic for the Process Update File – Local functionality is to determine if a file containing update data should be applied to the database or not. The following sequence diagram documents this logic.

Figure 17: Local Incremental database update process



Presentation Layer

The are no presentation layer classes for the Process Update File – Local functional area.

### PECS and DATUP Coordination for FDB Incremental process

#### System Features

This feature has been implemented in response to a problem discovered on the processes run by PECS and DATUP to utilize the same FDB Incremental files, here’s a description of such problem extracted from the PECS\_Comparison\_Report\_Process\_Issue\_Design\_Options document:

FDB Comparison Report is generated by PECS prior to DATUP applying the Drug Updates to the National Oracle Database. Delays on successful DATUP execution may be experienced; during this time the reports will be ahead of reality and vice versa if PECS process to download the file is not successful.

So the existing process introduces scheduling dependencies. PECS downloads the “FDB-DIF” incremental files from the SFTP Server, if this process fails due to Network issues (Router being down etc.) or the SFTP Server is down, the Comparison report will not be generated, if by the time DATUP runs (scheduled to run 1 hour after the PECS process) the above issue is resolved then the FDB database will be updated with FDB drug updates but PECS would not have make use of the file the way it intended to.

One of the suggestions made in the document to fix the issues described above involved the use of a JNDI flag in the WebLogic server in order to coordinate the processing of the FDB Incremental files between PECS and DATUP, such solution has been implemented on PECS 6.0 and DATUP 3.0, here’s an extract from the document (the actual implementation varied a bit from the proposed solution to make the process more efficient and less convoluted):

1. DATUP Enhancement:
   1. Check if FDB Drug Incremental updates available on the SFTP Server, if available download the update file.
   2. Set Global Flag(jndi) in WebLogic Domain for PECS – FDB incremental update available on WebLogic Server.
   3. Check the Global Flag after specified time (per timing parameter in configuration file) if PECS generated the “Comparison Report”. If PECS generated the report successfully process the Drug updates, if PECS did not generate Report-> generate an error/error message. (Message sent out to resolve the issue to pertinent resources).
2. PECS Enhancement:
   1. Check the Global flag in WebLogic Domain, at specified time per timing parameter (delayed by at least 20 minutes from DATUP scheduled time) in configuration file. If flag is changed, run the Comparison Report process and reset the flag.

#### Design Elements

JNDI Resource

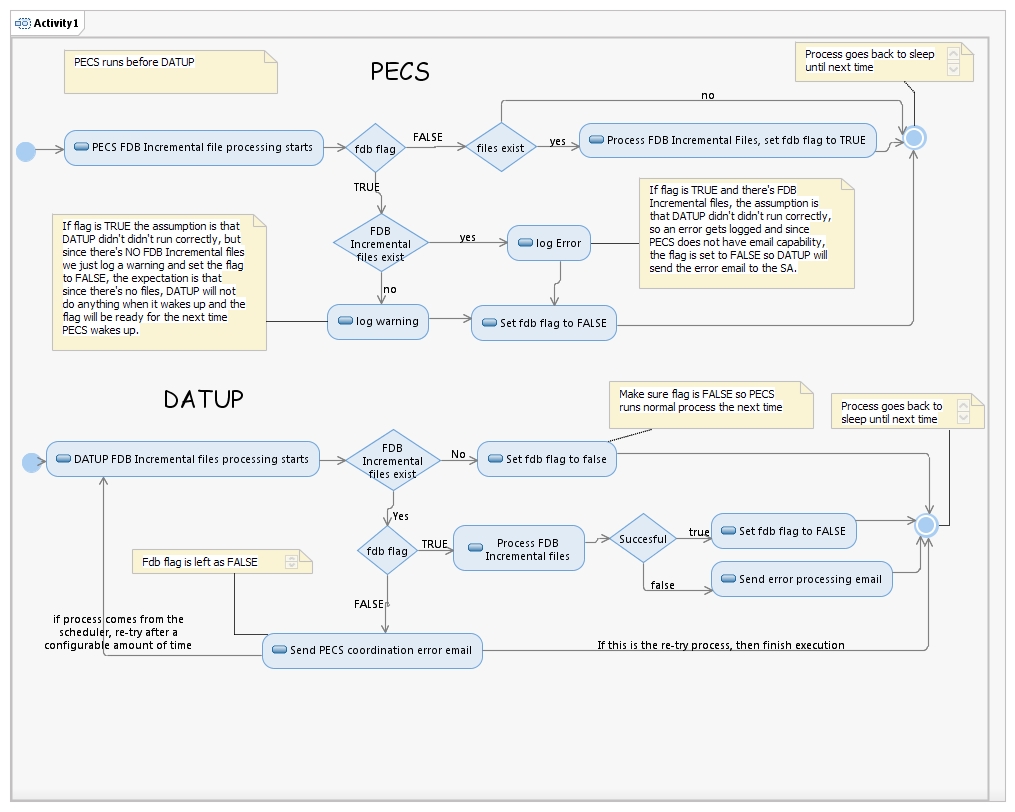
A new Boolean JNDI resource is created on WebLogic by DATUP right when it comes up, this resource is called fdb\_comparison\_report\_created\_flag and it lives right at the root of the DATUP’s WebLogic server JNDI tree. The initial value of it is FALSE.

This Boolean resource also known as fbg flag in the activity diagram on next section is updated by both PECS and DATUP to coordinate the access and processing of the FDB Incremental files in the SFTP server.

Business Layer

The following activity diagram describes the process that takes place between PECS and DATUP when it comes to the coordination between both apps for the use of the FDB Incremental files. The “Process FDB Incremental Files” activity for DATUP is described on the “Process for FDB Incremental database update” flow diagram in the Business Layer’s portion of Section 6.2.2 of this document. The “Process FDB Incremental Files” activity for PECS is found on section 6.4.2 of the PECS document.

Figure 18: Coordination activities between PECS and DATUP for FDB Incremental files processing



## Network Detailed Design

N/A

## Service Oriented Architecture / ESS Detailed Design

N/A

# External System Interface Design

The external interface design describes the interfaces between pre-existing systems and the system being developed. The DATUP application needs to externally interface with the legacy systems VistA Local (for authentication and authorization services). It will also interface with the DATUP application. The external interfaces are further described in the following sections and in the DATUP Interface Control Document (ICD).

## Interface Architecture

DATUP also interfaces with a SFTP Server to receive the custom update file that contains customizations performed in PECS and FDB Drug update file. The following is a high level view of the process. The figure below illustrates the logical system components for the National and Local environments. The National components are responsible for verifying and publishing FDB-DIF and FDB-Custom updates to the SFTP Server. The Local components then consume and apply the verified updates in an automated manner.

1. DATUP – Implements the FDB-DIF update business logic.
2. Scheduler – Background process for scheduling DATUP.
3. WebLogic – Application server environment for DATUP and PECS.
4. Configuration Files – Define the configuration settings for PECS and DATUP.
5. Email Templates – Template emails for notifications sent to National/Local Managers.
6. SFTP Server – SFTP Server that hosts the FDB-DIF update archives.
7. Email Server – Email relay server.
8. PECS – Implements the FDB-Custom drug business logic.
9. CT Staging Database – Stores PECS FDB-Custom modifications.
10. DATUP Database – Stores DATUP site update history.
11. FDB-DIF Database – Stores the FDB-DIF drug database.
12. Legacy VistA – Existing VistA Server for Security Interface.

DATUP deploys to both national and local instances. Both national and local instances of DATUP deploy to Oracle WebLogic. The PECS 6.0 production target involves systems hosted at both AITC and SDE.

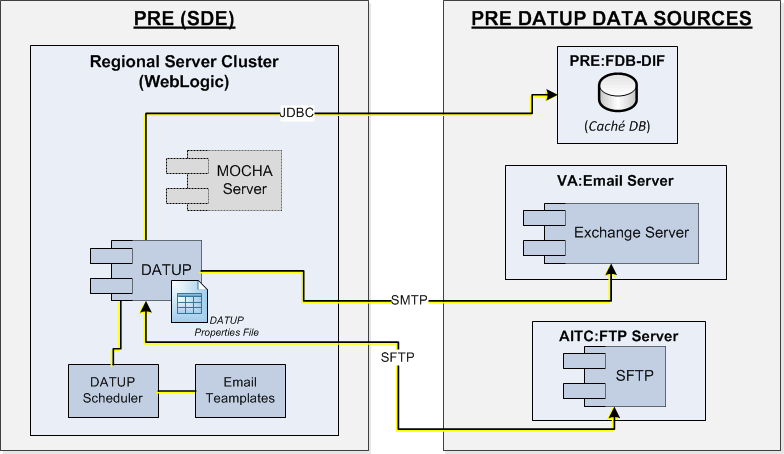
DATUP NATIONAL Deployment

The diagram in Figure 20 depicts the national deployment of DATUP. DATUP national shares data sources with DATUP local with the key difference being the target FDB-DIF is the Oracle relational database hosted in AITC. The national production target for DATUP 3.0 is WebLogic 12.1.1 running Java 1.7. At the national level, DATUP deploys to the same JVM as PECS eliminating the need for addition hardware but creating a dependency on the target deployment architecture.

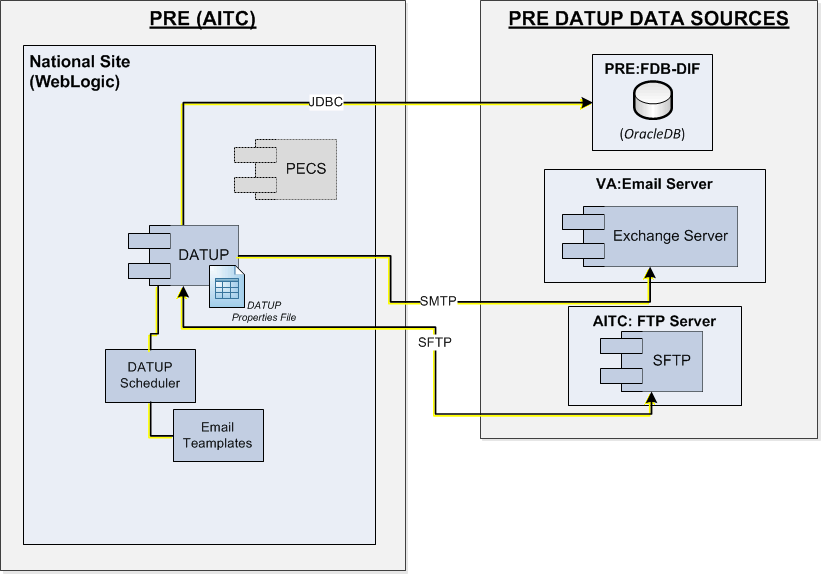
DATUP LOCAL Deployment

At the local level, DATUP deploys to the same RSC as MOCHA Server eliminating the need for addition hardware but creating a dependency on the target deployment architecture. The DATUP 3.0 local production target is WebLogic 12.1.1 running Java 1.7. Figure **19**, DATUP Local, shows the Local deployment.

**Figure 19: DATUP Local**



**Figure 20: DATUP National**



## Interface Detailed Design

Refer to PECS and MOCHA Server ICDs for DATUP Interface detailed design, as we did not roll out DATUP as a separate application. DATUP is a Common Service (J2EE) utility used by PECS/PPS-N so it does not require its own ICD/PAD.

Note: Due to policy constraints, active links cannot be included in this document. Please copy and paste the URLs into your browser.

PECS ICD: http://vaww.yourserver.domain/projects/pre/PRE\_IPT\_Rev/PRE\_IPT\_Rev\_PECS/Lists/Links/AllItems.aspx

MOCHA Server ICD:

http://vaww.yourserver.domain/projects/pre/PRE\_IPT\_Rev/MS1-1\_Review/default.aspx

# Human-Machine Interface

DATUP is a Java utility used to synchronize data in FDB DIF databases. DATUP exposes no Web Services to the MOCHA VistA application and has no Human Front-end GUI Interface. Therefore, the subsections under the Human-Machine Interface are not applicable.

# Security and Privacy

## Security

DATUP only security issue is the transfer of files containing update data. In order to be in line with VA Security Guidelines DATUP utilizes SFTP to transfer update data.

## Privacy

N/A

Attachment A – Approval Signatures

This section is used to document the approval of the System Design Document. The review should be conducted face to face where signatures can be obtained ‘live’ during the review. If unable to conduct a face-to-face meeting then it should be held via LiveMeeting and concurrence captured during the meeting. The Chair of the governing Integrated Project Team (IPT), Business Sponsor, IT Program Manager, and Project Manager are required to sign.

▲Signed:

*, Program Manager, Pharmacy Reengineering*

*Integrated Project Team (IPT) Chair & IT Program Manager*

▲Signed:

*, PBM, Director, Clinical Informatics/Reengineering*

*IPT Member & Business Sponsor*

▲Signed:

*, Project Manager, Pharmacy Reengineering (PECS)*

▲Signed:

*, Program Manager, SDE PAO*

*IPT Member & SDE Representative*

1. Additional Information
   1. RTM

Please refer to “DATUP Requirements Traceability Matrix.xlsx” artifact for Comprehensive Requirements Traceability Matrix. (however a high level table below has RTM for reference purpose only).

This section provides the traceability from requirements. Only requirements that trace to external services are included in this document.

Table 63: Definitions for Mapping SRS to Interfaces

| Attribute | Description |
| --- | --- |
| Requirement | The requirement unique identifier and requirement’s text that references a specific requirement. |
| Interface | The name of the interface that implements or accesses the required functionality. |
| Service(s) | The DATUP service that implements or accesses the required functionality. |
| Method(s) | The method that implements or fulfills the required functionality. The methods are exposed DATUP methods. |

Table 15 shows how individual requirements are traced to interfaces, services, and the related methods that satisfy each requirement.

Table 64: Requirement Traceability

| Requirement | Interface | Service(s) | Method(s) |
| --- | --- | --- | --- |
| SPEC1196: FDB-DIF incremental files are processed by the date sequences, from the oldest to newest if more than one file is available for processing. | NationalDifUpdateCapability  LocalDifUpdateCapability | DifUpdateCapabilityImpl | sortByDependencyOrder |
| SPEC1197: Order Check Validation is done before and after the automated data update to ensure integrity of FDB-DIF Drug database. | LocalDifUpdateCapability | LocalDifUpdateCapability | Execute |
| SPEC1198: The application should automatically check daily for available FDB-DIF update files received from CMOP files to process. | DifUpdateScheduler | Scheduler | scheduleNextTimer |
| SPEC1199: The application should automatically check daily for available VA Custom Incremental files to process. | NationalDifUpdateCapability  LocalDifUpdateCapability | DifUpdateCapabilityImpl | retrieveFdbArchive |
| SPEC1200: The application should be configurable, for each installation, to begin the update process at a configured time if the FDB and/or VA incremental updates are present. | Configuration | Scheduler | SCHEDULED\_TIME |
| SPEC1201: Validation of Success or Failure of the Updates | NationalDifUpdateCapability  LocalDifUpdateCapability | VerificationTester | executeOrderCheckTests |
| **SPEC1202:** The application should provide automated email notifications of success or failure of the update. The email addresses group for these notifications is configurable. | NationalDifUpdateCapability  LocalDifUpdateCapability | NationalDifUpdateCapability  LocalDifUpdateCapability | notifyManagers |
| **SPEC1203:** The application should be able to manage the FDB-DIF and VA custom files stored nationally and at the region or local sites. | NationalDifUpdateCapability  LocalDifUpdateCapability | NationalDifUpdateCapability  LocalDifUpdateCapability | Execute |
| **SPEC1204:** The application should provide web based compliance reporting of all sites; | Reporting | Reporting | get |
| **SPEC1205:** The application should be able to upload Images that are received with the weekly FDB-DIF updates to the PECS/PPSN Production Server (National). | NationalDifUpdateCapability | NationalDifUpdateCapability | processImageFile |

* 1. Packaging and Installation

DATUP software deployment deliverables are packaged with a Version Description Document (VDD) that details the software delivered, including version numbers. DATUP Installation Guide is also delivered, detailing how the software should be installed.

* 1. Design Metrics

The DATUP utility and deliverables follow the VA defined Quality and SQA process for review, a list of minor and major defects are generated, which are subsequently acted upon to rectify them.

* 1. Required Technical Documents

The following documents must be submitted for review to support proper approval:

• Product Architecture Document;

• Disaster Recovery Plan;

• Interface Data Mapping

• Conformance Validation Statement (CVS) - Section 508

For additional information regarding how to obtain proper approval for this project, refer to the following documents:

Note: Due to policy constraints, active links cannot be included in this document. Please copy and paste the URLs into your browser.

• IT Infrastructure Standards

http://vaww.eie.domain/techstrategy/ET/FieldSolutionDOCs/Forms/AllItems.aspx?RootFolder=%2ftechstrategy%2fET%2fFieldSolutionDOCs%2f1%2fQuickBrief%20Books&FolderCTID=&View=%7b14A17365%2dDE2C%2d4A54%2dA55A%2d75BF03B49A34%7d

• Systems Engineering and Design Review (SEDR) process

http://vaww.eie.domain/techstrategy/TAR/TAR%20PROCESS/Forms/AllItems.aspx

• Enterprise Architecture Web page

http://vaww.server.domain/

• One-ONE-VA TRM Approved Technologies/Standards List

http://trm.oit.domain/TRMHomePage.