Veterans Personal Finance System (VPFS)

Systems Management Guide

Version 1.2.0

July 2020

Department of Veterans Affairs (VA)
Office of Information and Technology (OIT)
Enterprise Program Management Office (EPMO)

Revision History

The following table displays the revision history for this document. Revisions to the documentation are based on continuous dialogue with the VPFS development team.

Date	Revision	Description	Author
12/1/06	1.0	Initial release of document	Project Manager: Richard LeBlanc Developers: Mark Enfinger Stefan Foerster
06/03/10	1.1.2	Page 5 – Modified EVS to EPS. Page 9, 14 -- Updated SDS, VLJ and KAAJEE versions. Page 16 – Updated Data Base version. Section 4.6 – Replaced PATS with VPFS. Page 5,7,26- Replaced EMC with EIE Paragraph 5.12.1- added the missing table „std_facilitytype‟ These changes are associated with the VPFS patch release 1.1.2	Ram Gade
06/28/10 12/22/10	1.1.3	Page 9, 14 -- Updated SDS version from 13 to 18. Page 9 –Added Java version These changes are associated with the VPFS patch release 1.1.3	Sowjanya (Ram) Gade
From 01/05/2019 – 02/05/2020	1.2.0	Software & Plug-in updates	Developers: Leena Nathan Ralph Katieb Jonathon Geffen Hean Gau Justin Loverme

Table of Contents

iii
Table of Contents

1 Introduction

Veterans Personal Finance System (VPFS) is the reengineered version of VistA Personal Funds of Patients (PFOP) system, also known as Integrated Patient Funds (IPF), the mini-banking system that manages the accounts of patients in the VA hospital system.

The VPFS Systems Management Guide gives a technical overview of VPFS for supporting and maintaining the application. The intended audience of this guide is: Information Resource Management (IRM), Enterprise Infrastructure Engineering (EIE) Health Solutions, and Enterprise Product Support (EPS).

VPFS is a web-based application. Both the VPFS application and database are maintained at the EIE in Falling Waters, VA.

1.1 Chapter Descriptions and Roles

The guide is divided into the following sections:

Chapter	Chapter Name	Intended Audience
1	Introduction	Java developers and System Admin
2	System Requirements	Java developers and System Admin
3	VPFS Application	Java developers and System Admin
3	Business Rules Implementation	Business layer developer at EIE, Maintenance and EPS Support
4	Database – M VistA	IRM staff, Maintenance, and EPS Support
5	Database – Oracle	Database Administrator (DBA) at the EIE, Maintenance, and EPS Support
6	Integration Agreements	EIE, Maintenance, and EPS Support
7	Patient Demographics Updates	EIE, Maintenance, and EPS Support

8	Data Migration	EIE and EPS Support
9	VPFS Troubleshooting	EIE and EPS Support
10	VistAMigrate Troubleshooting	EIE and EPS Support

This page is left blank intentionally.

2 System Requirements

2.1 Server Environment

The following components are required for the production VPFS environment:

- Oracle 12g Server
- WebLogic Server 10.3.6 (or later)
- VistA Server

Additional components such as a hardware load balancer and Secure Sockets Layer (SSL) server are being managed by the EIE.

2.1.1 Oracle Server

The Oracle server must have Oracle 10g Server installed.

VPFS requires the following schemas installed in an Oracle instance accessible by the WebLogic server:

- VPFS – Primary schema used by VPFS. All VPFS data is stored in this schema.
- VistAMigrate – Required to perform migrations from PFOP. After all stations have been migrated, the VistAMigrate schema can be removed.
- SDS – The VPFS schema has a view that references certain Standard Data Service (SDS) tables, therefore this schema must be accessible from the VPFS schema.
- KAAJEE – The KAAJEE schema is used by the Kernel Authentication & Authorization for J2EE (KAAJEE) Security Service Provider Interfaces (SSPI), not VPFS directly. Typically, it would be installed in the same database instance as VPFS, although technically, it does not have to be.

2.1.2 WebLogic Server

WebLogic 10.3.6 was used during development and has been identified as the target production version.

All dependencies must be installed and configured in the WebLogic domain designated for VPFS. This includes VistALink, KAAJEE SSPIs, Person Service Lookup (PSL), and Person Service Construct (PSC). (Electronic Signature does not have any installable components on the application server.) Refer to the documentation for each dependency for any specific requirements for that dependency.

Database connection pools and datasources must be installed for VPFS, VistAMigrate, and SDS.

Note: This version of VPFS uses VistALink libraries that have reverse-IP check enabled. Confirm all entries in the VistALink connectorConfig.xml file have the reverse IP information configured. Otherwise VistALink connector will not be able to connect out to those systems. This was done for security purposes and to pass Fortify compliance.

2.2 Developer Environment

VPFS Java Enterprise developer workstations are dependent on remote and local services. All VA Java service dependencies are deployed locally. Oracle databases were available both locally and remotely. VistA databases were utilized remotely.

2.2.1 Development Platform

VPFS was developed using the following infrastructure on Windows workstations:

7

Application	Description
WebLogic 10.3.6	All service dependencies are required to be setup on a local WebLogic server.
Oracle 12g	Individual development and test database installed locally
InterSystems Caché	Individual development and test VistA database installed locally *(for M developers)*

2.2.2 Development Tools

The following tools were used for VPFS development:

Application	Description
Eclipse 3.1 and MyEclipse 4.0	Integration of Eclipse and WebLogic 10.3.6 allowed hot deployment and hotswap debugging.
Ant 1.6	Automated build tool used for VPFS and VistAMigrate. Each project root directory contains a build file named build.xml.
Microsoft Visual Source Safe	Source Control Management
Quest Software TOAD 8.0	TOAD was used to connect to the Oracle database(s) to verify that data was updated correctly. TOAD can also be used to generate database scripts from a developed schema, and has a PL/SQL editor for editing stored procedures, triggers, etc.
Oracle Enterprise Manager Console	Installed with Oracle 10g Server. Allows DBA to manage Oracle database instances.
SmarTerm	Terminal emulation software for connecting to remote VistA servers. *(for M developers)*

2.2.3 Frameworks and Libraries

2.2.3.1 Development Frameworks

The following development frameworks are used in the VPFS application:

- Log4J – Logging framework
- Struts – Servlet framework implementing a variation of the classic Model-View-Controller (MVC) design paradigm
- Tiles – JavaServer Pages (JSP) templating framework

2.2.3.2 Additional Libraries

All libraries required for building VPFS are included in the project lib directory except the following system libraries, included with WebLogic:

Table of Contents

Library Name	Description
J2EE v 1.7_80	Java Enterprise Edition, Java Development Kit
JRE System Library 1.8	General Java Runtime Environment (JRE) class library
WebLogic 10.3.6	WebLogic tools, drivers, etc.

2.2.4 VA Services

The following VA services are deployed locally in a development Weblogic server:

- Standard Data Service (SDS) API jars 18.0
- VistALink for Java (VLJ) 1.6
- Kernel Authentication Authorization Java Enterprise Environment (KAAJEE) 1.0.1.003
- Person Service Lookup (PSL) 4.0.4.3
- Patient Service Construct (PSC) 2.0.0.8
- Electronic Signature (eSig) 1.0.0.024

2.2.5 Builds

Ant is used as the build tool for the VPFS and VistAMigrate applications, as well as to assemble the database script zip file and the release package.

Each project has its own build file (build.xml) containing *targets*, or different actions to take on that project. For example, the *ear* target in the VPFS build.xml will compile the application, prepare the application for WebLogic deployment, and update the build date, all before creating the ear file.

2.2.5.1 VPFS Build Targets

Target	Description

build	Compiles the application source code, prepares the application for WebLogic deployment (calls weblogic.appc), and updates the build date in the Version.properties file.
clean	Removes all compiled files to ensure all classes are recompiled.
ear	Builds the application prior to creating the application ear file.
javadoc	Generates javadoc for all VPFS classes (gov.va.med.vpfs.* package).
release	Performs a clean rebuild of the application, creates the application ear file, generates the versioned readme and Release Notes from the templates, then generates a MD5 checksum for the ear file.
verifychecksum	Verifies the MD5 checksum for the application ear to ensure the ear file has not changed since the checksum was generated.
version	Increments the build number in the Version.properties file.

2.2.5.2 VistAMigrate Build Targets

Target	Description
build	Compiles the application source code, prepares the application for WebLogic deployment (calls weblogic.appc), and updates the build date in the Version.properties file.
clean	Removes all compiled files to ensure all classes are recompiled.
ear	Builds the application prior to creating the application ear file.
release	Performs a clean rebuild of the application, creates the application ear file, generates the versioned readme and Release Notes from the templates, then generates a MD5 checksum for the ear file.
verifychecksum	Verifies the MD5 checksum for the application ear to ensure the ear file has not changed since the checksum was generated.
version	Increments the build number in the Version.properties file.

3 VPFS Application

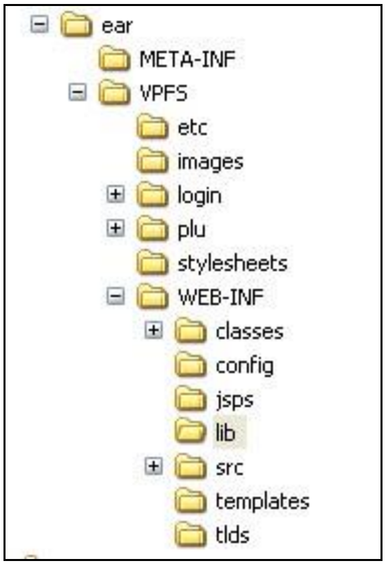
VPFS is designed to run on a Java Enterprise Edition (J2EE) 1.3 compliant application server running on a 1.4 Java Virtual Machine (JVM).

VPFS can be deployed in two basic ways on Weblogic: an

- Enterprise Archive (.ear) file, or
- an “exploded” ear directory structure

3.1 Enterprise Archive

VPFS deploys as a Java Enterprise Archive named VPFS-*major.minor.revision.build*.ear containing the following elements.



```
⊟ ear
    META-INF
  ⊟ VPFS
      etc
      images
    ⊞ login
    ⊞ plu
      stylesheets
    ⊟ WEB-INF
      ⊞ classes
        config
        jsps
        lib
      ⊞ src
        templates
        tlds
```

3.1.1 Application.xml

This file exists in the META-INF directory per the J2EE 1.3 specifications. The contents are the modules and logical security roles of the VPFS application.

3.1.2 Weblogic-application.xml

This file exists in the META-INF directory. No special settings in this deployment descriptor

3.1.3 Web Application

The VPFS enterprise archive contains the VPFS web application.

The VPFS directory is the web context root. The web context root contains the following application directories: (The files in **bold** text contain elements that are configurable during deployment.)

Directory	Description

Directory	Description
etc	Contains common cascading stylesheets (CSS) and JavaScript files for VPFS
images	Contains application images (gif and jpeg files) for VPFS.
login	Contains KAAJEE login pages. These pages are used to authenticate users.
plu	Contains PSL pages. These pages are used to look-up and register patients in VPFS.
stylesheets	Contains common CSS stylesheets for PSL (not VPFS).
WEB-INF	Contains web.xml, **weblogic.xml**, and various configuration files, taglibs, classes, JSPs, and libraries per the Servlet and JSP specifications.
WEB-INF/classes	Contains the configuration property files for PSL (**PatientLookup.properties**) and PSC (**PatSvcPkg.properties**).
WEB-INF/config	Contains KAAJEE (**kaajeeConfig.xml**) and log4j (**log4j.xml**) configuration files. Also contains application configuration files for Struts, Tiles, etc. Any modification of these files shall be considered a change to application code and should be managed accordingly.
WEB-INF/jsps	Contains the JSP pages for VPFS.
WEB-INF/lib	Contains the external jar files used by the web application.
WEB-INF/templates	Contains the JSP Tiles for VPFS.
WEB-INF/tlds	Contains the tag library descriptors.

3.1.3.1 Web.xml and Weblogic.xml

Web.xml is the main configuration file for the VPFS web application. Web.xml contains:

- context parameters servlets
- servlet mappings
- security roles
- security constraints
- login configuration
- taglib declarations
- welcome files
- error pages, mime mappings, listeners, and environment entries

Weblogic.xml contains the mapping of the application role to the actual WebLogic security realm. It also contains the security role assignment for the KAAJEE LoginController.

3.1.3.2 PSL and PSC Configuration Files

The PatientLookup.properties and PatSvcPkg.properties files are the configuration files for PSL and PSC, respectively, located in the WEB-INF/classes directory. These files contain the providerURL, securityProvider and securityCredentials properties that must be updated to reference the correct location and security information for connecting to the PSL and PSC Enterprise JavaBeans (EJBs).

Refer to PSL and PSC documentation for more information about the contents of these files.

3.1.3.3 KAAJEE Configuration File

The kaajeeConfig.xml file is the configuration file for KAAJEE, located in the WEBINF/config directory. This file contains the list of station numbers that will appear in the Institution drop-down list on the KAAJEE login page.

Refer to KAAJEE documentation for more information about the contents of this file.

3.1.3.4 Other Configuration Files

VPFS uses the Struts framework with the Tiles templating framework internally. The struts-config.xml and tiles-defs.xml files for configuring these frameworks are located in the WEB-INF/config folder.

3.1.3.5 Stylesheets

The main CSS stylesheet for VPFS is vpfs.css, located in the /etc directory. The main CSS stylesheet for the text-only (508-compliant) mode is vpfs508.css, also located in the /etc directory. These stylesheets are used to maintain a consistent look and feel throughout VPFS.

3.1.3.6 Libraries

The following libraries define application programming interfaces (APIs) that are included and required by VPFS, and are located in the WEB-INF/lib directory:

Library Name	Description
CAIP.jar	Used by PSL/PSC
commons-beanutils.jar	Jakarta Commons library, used by Struts/Tiles
commons-collections-3.1.jar	Jakarta Commons library, used by Struts/Tiles
commons-dbcp-1.2.1.jar	Jakarta Commons library
commons-digester.jar	Jakarta Commons library, used by Struts/Tiles
commons-fileupload.jar	Jakarta Commons library, used by Struts/Tiles

commons-httpclient-2.0.1.jar	Jakarta Commons library
commons-lang-2.0.jar	Jakarta Commons library
commons-lang.jar	Jakarta Commons library
Library Name	**Description**
commons-logging-1.0.3.jar	Jakarta Commons library
commons-pool-1.2.jar	Jakarta Commons library
commons-resources.jar	Jakarta Commons library
commons-services.jar	Jakarta Commons library
commons-validator.jar	Jakarta Commons library, used by Struts/Tiles
dbunit-2.0.jar	
displaytag-1.0-b2.jar	
esig-1.0.0.024.jar	Electronic Signature
jakarta-oro.jar	Jakarta Commons library, used by Struts/Tiles
jakarta-regexp-1.3.jar	Jakarta Commons library
jaxen-core.jar	XML (eXtensible Markup Language) parsing API
jaxen-dom.jar	XML parsing API
jstl.jar	JSTL (Java Standard Template Library)
junit-3.8.1.jar	Unit Testing API
kaajee-1.0.1.003.jar	KAAJEE
local_policy.jar	
log4j-app.2.10.0.jar	Log4j
log4j-core.2.10.0.jar	Log4j
LongList.jar	Used by PSL
NamingDirectoryService-client.jar	Used by PSL/PSC
PatientServiceR2.jar	PSC
pslWeb_4.0.4.3.jar	PSL
standard.jar	JSTL
struts.jar	Struts

US_export_policy.jar	
vha-stddata-basic-18.0.jar	SDS
vha-stddata-client-18.0.jar	SDS
vha-stddata.jar	SDS
vpfsTags.jar	
wlclient.jar	Standard J2EE classes

3.1.3.7 Classes

The class files for the VPFS web archive are located by package under the WEBINF/classes directory. The class packages are described in the following table.

Table of Contents

Package Name	Description
gov.va.med.vpfs.action	Action handler (Servlet *controller*) classes called by Struts.
gov.va.med.vpfs.biz	Business logic classes; perform non-form-based data validation, manage calls to the database access layer.
gov.va.med.vpfs.dao	Data access objects; classes and interfaces for reading from or writing to the database.
gov.va.med.vpfs.jsp	JSP-specific classes used by Tiles to build menu items.
gov.va.med.vpfs.tag	VPFS taglib, custom controls.
gov.va.med.vpfs.util	Common utility classes: formatters, service locators, etc.
gov.va.med.vpfs.vo	Value object classes; basic data objects.

3.2 HealtheVet Configuration Files

VPFS depends on the following service dependency configuration files that may need updating during deployment. Refer to the documentation for these dependencies for more information about the configuration properties. These files are currently packaged in the VPFS application enterprise archive (ear) file, however may alternatively reside in the HealtheVet common configuration location:

- kaajeeConfig.xml – a KAAJEE configuration file (must remain in the ear file)
- PatientLookup.properties – a PSL configuration file
- PersonLookupResources.properties – a PSL configuration file
- PatSvcPkg.properties – a PSC configuration file

These files are installed on the WebLogic server in the installation directory for the specified component:

- KaajeeDatabase.properties – a KAAJEE configuration file (SSPI)
- gov.va.med.vistalink.connectorConfig.xml – a VLJ configuration file

3.3 Other Configuration Properties

There are a number of configuration properties for VPFS in the VSYSTEM_PARAMETER table. Changes to these properties should be made with care and a full understanding of how the parameter effects the VPFS application.

Parameter Name	Description	Default Value
APPLICATION_PROXY_USER	VistALink application proxy user account, required for PSC to be able to connect to the VistA sites.	VPFS,APPLICATION PROXY
DB_VERSION	**Internal.** Must match database version in web.xml.	1.0.5.001
EMAIL_ENABLED	Set whether email notifications of override conditions are enabled. Allowed values are “true” and “false”. If true, SMTP_SERVER must also be specified.	false
PATIENTUPDATE_SCHEDULE_TIME	Start time of the scheduled update process.	1:00 am
PRODUCTION_MODE	**Internal.**	true
SMTP_SERVER	DNS hostname or IP address of the SMTP server (to send email).	smtp.va.gov
VISTALINK_ACTIVE	**Internal.**	true

3.4 Logging

Log4j is the logging framework used in VPFS. Logging is configured via the log4j.xml configuration file located in the Health*e*Vet common configuration location, specified for each server using the -Dlog4j.configuration JVM argument. Log4j defines the concepts of “appenders” that define individual log files and the layout of log data in those files, and “loggers” that define what log types and levels are sent to the various appenders. Refer to log4j documentation for more information about the contents of this file.

A sample log4j configuration file (log4j_sample.xml) is provided in the VPFS distribution package.

The default log file location is the log directory. This directory is created under the Node Manager root when running on a managed server (e.g. /u0*n*/app/bea/weblogic81/common/nodemanager/log/). When running on an admin server (not recommended for production environments!), this directory is created under the domain root (e.g. /u0*n*/app/bea/user_projects/domains/domVpfs/log/).

3.4.1 Application Server Log File

WebLogic creates various log files for each admin or managed server started. Log4j can be configured to write to these files by sending log to the Console appender. The location and name of these log files is dependent on the type of server:

- Managed Servers – There are 2 log files created for each managed server. These files are located in a NodeManagerLogs/<*domain-name*>_<*server_name*> directory under the Node Manager root.

- Administration Server – The admin server log is *<server-name>*.log, located in the domain root directory.

The application server log is configured as the root logger in the default VPFS log4j configuration, meaning any uncaught exceptions or errors will be written to this log.

Package	**Level**	**Description**
gov.va.med.term.access	Info	SDS
gov.va.med.term.access. maint.messaging.hl7	Info	SDS
gov.va.med.vistalink	Info	VistALink*
gov.va.med.authentication	Info	KAAJEE

*VistALink has its own log4j configuration file that may record VistALink log information in another location. Refer to VistALink documentation for more information.

3.4.2 VPFS Log File

The default VPFS log file is vpfs.log. Any VPFS generated errors or log messages will be written to this log.

Package	**Level**	**Description**
gov.va.med.vpfs.action	Info	User Interface exception handler
gov.va.med.vpfs.biz	Info	Business layer exception handler
gov.va.med.vpfs.dao	Info	Data access exception handler
gov.va.med.vpfs	Info	General VPFS exception handler

3.4.3 PSL Log Files

There are two standard log files configured for PSL: psl.log for general PSL errors or log messages, pslmetrics.log for PSL metrics.

Package	**Level**	**Description**
gov.va.med.person.lookup	Info	PSL exception handler, writes to general PSL log
gov.va.med.person.lookup.common. PLUMetricsManager	Info	PSL Metrics handler, writes to metrics PSL log

3.4.4 PSC Log File

The standard PSC log file is PatSvc.log. Any PSC generated errors or log messages will be written to this log.

Package	**Level**	**Description**

gov.va.med.patientadmin	Info	PSC exception handler

3.4.5 VistALink Log File

The standard VistALink log file is vlj.log. Any VistALink generated errors or log messages will be written to this log.

Package	Level	Description
gov.va.med.patientadmin	Info	PSC exception handler

3.5 Exceptions

VPFS internal checked exceptions are described in Javadoc:

- gov.va.med.vpfs.util.VPFSException
- gov.va.med.vpfs.util.InvalidInstitutionException
- gov.va.med.vpfs.util.ServiceLocatorException
- gov.va.med.vpfs.util.VPFSESigServiceException
- gov.va.med.vpfs.util.VPFSSecurityException
- gov.va.med.vpfs.util.VPFSSystemException
- gov.va.med.vpfs.util.VPFSVistaLinkException
- gov.va.med.vpfs.tag.VPFSTagException − gov.va.med.vpfs.biz.VPFSBDException
- gov.va.med.vpfs.biz.EmailSendingException
- gov.va.med.vpfs.biz.EventLoggingException
- gov.va.med.vpfs.biz.InvalidAccountBalanceException
- gov.va.med.vpfs.biz.InvalidESignatureException
- gov.va.med.vpfs.biz.InvalidTransactionChecksumException
- gov.va.med.vpfs.biz.PatientDemographicsException
- gov.va.med.vpfs.biz.SystemUnavailableException
- gov.va.med.vpfs.biz.TransactionOverrideException −

  gov.va.med.vpfs.dao.VPFSDBException
- gov.va.med.vpfs.dao.DuplicateKeyException
- gov.va.med.vpfs.dao.ForeignKeyViolationException
- gov.va.med.vpfs.dao.RecordNotFoundException
- gov.va.med.vpfs.dao.SystemDBException

3.6 Service Imports

VPFS imports the following classes from VA services:

Standard Data Service (SDS)

gov.va.med.term.access.Institution

gov.va.med.term.access.InstitutionCode

Electronic Signature

gov.va.med.esig.utilities.ESigEncryption

gov.va.med.esig.utilities.ESigException

gov.va.med.esig.utilities.ESigValidation

gov.va.med.esig.utilities.ESigDataAccess

KAAJEE

gov.va.med.authentication.kernel.LoginUserInfoVO

VistALink for Java

gov.va.med.vistalink.adapter.cci.VistaLinkConnection

gov.va.med.vistalink.adapter.cci.VistaLinkConnectionFactory

gov.va.med.vistalink.adapter.cci.VistaLinkDuzConnectionSpec

gov.va.med.exception.FoundationsException

gov.va.med.vistalink.institution.InstitutionMapNotInitializedException

gov.va.med.vistalink.institution.InstitutionMappingDelegate

gov.va.med.vistalink.institution.InstitutionMappingNotFoundException

Patient Service Lookup (PSL)

gov.va.med.person.lookup.common.IPLUConstants

gov.va.med.person.lookup.patient.transfer.PatientLookupBean

gov.va.med.person.lookup.patient.transfer.UserBean

gov.va.med.person.lookup.provider.exception.ProviderLookupDelegateException

gov.va.med.person.lookup.provider.transfer.ProviderLookupCriteria

gov.va.med.person.lookup.provider.transfer.ProviderVO

gov.va.med.person.lookup.transfer.LookupCredential

gov.va.med.person.lookup.ui.web.common.PLSessionMgr

Patient Service Construct (PSC)

gov.va.med.patientadmin.common.AddressType

gov.va.med.patientadmin.common.IAddress

gov.va.med.patientadmin.common.IIdentifier

gov.va.med.patientadmin.common.IPatientId

gov.va.med.patientadmin.common.IVistaDate gov.va.med.patientadmin.common.PatientId

gov.va.med.patientadmin.common.RequestType

gov.va.med.patientadmin.common.RequestedServices

gov.va.med.patientadmin.delegate.IPatientServiceDelegate

gov.va.med.patientadmin.delegate.IPatientServiceRequest

gov.va.med.patientadmin.delegate.PatientServiceDelegate

gov.va.med.patientadmin.delegate.PatientServiceRequest

gov.va.med.patientadmin.exception.PatientServiceException

gov.va.med.patientadmin.transfer.IAddressDemographicsTO

gov.va.med.patientadmin.transfer.IADTTO

gov.va.med.patientadmin.transfer.IEnrollmentEligibilityTO

gov.va.med.patientadmin.transfer.IIncompetenceInformationTO

gov.va.med.patientadmin.transfer.IPatientServiceTO

gov.va.med.patientadmin.transfer.IPrimaryDemographicsTO

3.7 Source Code

Source code for VPFS is available in the VPFS SourceSafe database.

3.8 Javadoc

Javadoc for VPFS is available in the VPFS SourceSafe database.

3.9 Business Rules Implementation

3.9.1 Security

Application security is managed via constraints placed on the URLs that a user may access in the web.xml configuration file. Security keys assigned to users in VistA are retrieved by KAAJEE and mapped to roles by the application server. A list of VPFS security keys is available in the *VPFS User Guide* and in Section 4.9 of this document.

Programmatic security is used for hiding invalid options on individual pages. The tags defined in the VPFS tag library are all role-aware. Availability is determined by checking if the user has one of the roles specified in the tag roles attribute by calling request.isUserInRole().

3.9.2 Transactions

User transactions are used when updating the database. The JTAUtil class provides methods to begin, commit, or rollback transactions. The Business Delegate layer manages the transaction context.

VPFS uses Oracle transactions in those stored procedures triggered by a scheduled job or trigger.

3.9.3 Email Notifications

Notifications are sent via email when a transaction override condition occurs (in TransactionBD). Email settings are configured in the Maintain System Parameters page in VPFS (EMAIL_ENABLED and

SMTP_SERVER parameters) and the BDResources.properties file (WEBINF/src/gov/va/med/vpfs/biz/BDResources.properties).

3.9.4 Table Maintenance

Many of the drop-down lists in VPFS are configurable via the Administration options in VPFS. The income source, form, remark, payee, and reference types are configurable at the station level, while patient status, patient type, and payment type are configurable at the national level.

3.9.5 Electronic Signature

Electronic Signatures are used to authenticate the user when posting transactions or editing a transaction deferral date. VistA serves as the authoritative source of electronic signatures. VPFS does not store the electronic signature, but simply uses the Electronic Signature API to verify the electronic signature entered by the user. VPFS also allows the user to change their electronic signature via the Electronic Signature API.s

4 Database - M VistA

Prior to using the VPFS or VistAMigrate application at a given site, the site must install two KIDS builds PRPF*3.0*15 and PRPF*3.0*16 that are associated with the Patient Funds package. These KIDS builds bring in routines, options, remote procedure calls (RPCs), and security keys. No new data files are brought in with this installation, and the existing Patient Funds files, routines, options and security keys are not changed by these patch installations. See the patch descriptions for instructions on the installation and setup.

The objective of PRPF*3.0*15 (Diagnostic Patch) is to allow sites to diagnose and resolve as many issues as possible before they migrate from VistA legacy Personal Funds of Patients (PFOP) to the Veterans Personal Funds System (VPFS). Install and use Diagnostic Patch 15 well in advance of your actual data migration date to allow time to run the diagnostic routine, review the reports, and make corrections in the M environment repeatedly to narrow down the count of errors.

Migration Patch PRPF*3.0*16 is used during the migration of data from PFOP to VPFS. This patch includes the following Remote Procedure Calls (RPC) functionality that enables the VPFS migration application to transfer specific Patient Funds legacy data to the reengineered VPFS system. This new functionality is not added to the legacy Patient Funds menu because this functionality is intended for use as an RPC only. This RPC is invoked by VistALink from an outside migration server. Proxy User for Nightly Updates: This patch uses a post install routine PRPFMR2 that will create a proxy user account for the VPFS application to use during nightly updates with all legacy systems. The post install results create entries in the Patch 16 installation report. The patch installation report will always contain one of three possible responses from the proxy account install process:

1. THE VPFS APPLICATION PROXY USER ACCOUNT HAS BEEN SUCCESSFULLY CREATED!
    - The account was created and no further action is required.
2. WARNING: THE VPFS APPLICATION PROXY USER ACCOUNT ALREADY PRESENT!!
    - The account was already present and the reason for this should be known by IRM. If there are any questions as to why the account is already present it should be investigated.
3. ERROR: THE VPFS APPLICATION PROXY USER ACCOUNT COULD NOT BE CREATED!!!
    - There was a problem and IRM should be approached for a resolution and patch 16 should be re-installed.

The purpose of patch PRPF*3.0*17 is to partially disable the legacy Patient Funds package. This patch is -ONLY- intended for installation in a VistA environment that has successfully completed the VPFS migration process for the Patient Funds package. If migration has been completed and the site is ready to DISABLE all standard user write or delete privileges to their legacy VistA Patient Funds package then proceed with installation of this patch.

All Patient Funds options and functionality that either adds, updates, or deletes Patient Funds data will be disabled for standard VistA users (Guardian address edit will still function normally). All other Patient Funds menu options, such as reporting, will continue to function normally and it is intended that the legacy VistA Patient Funds package still remain available for historical reporting purposes indefinitely.

4.1 Namespace

Routines, options, and remote procedures beginning with PRPF are used to migrate legacy data into VPFS. Routines, remote procedures and security keys beginning with PRFP are used during normal operation of the VPFS application.

4.2 Routines

Patch	Routine Name	Description
PRPF*3.0*15	PRPFDR1	This routine is used to report diagnostic information related to Patient Funds data.
PRPF*3.0*15	PRPFDR2	This is the main entry routine that is used to report diagnostic information related to Patient Funds data.
PRPF*3.0*15	PRPFDR3	This routine is used to report diagnostic information related to Patient Funds data.
PRPF*3.0*15	PRPFDR4	This routine is used to report diagnostic information related to Patient Funds data.
PRPF*3.0*15	PRPFDR5	This routine is used to report diagnostic information related to Patient Funds data.
PRPF*3.0*15	PRPFDR6	This routine is used to report diagnostic information related to Patient Funds data.
PRPF*3.0*16	PRPFMR1	This routine is used to extract and move Patient funds data to the migration server.
PRPF*3.0*16	PRPFMR2	This routine is used to create the VPFS proxy user for nightly updates.
PRPF*3.0*17	PRPFMR3	This routine is used to disable ADD, EDIT, or Delete privileges for all Patient Funds users.

4.3 Temporary Globals

The ^TMP global is used to store information during the data migration process. Each time the user runs either the Data Diagnostic or the Data Migration the data contained in ^TMP is purged. If the data does not get purged, there is also a purge all at the beginning of each process to ensure that no previous data is present.

Temporary Global Name	Description
^TMP("PRPF_DIAGX",$J,0)	Holds the purge date and the date the Diagnostic was last performed.
^TMP("PRPF_DIAGX",$J,PFSTAID,*error#*	Holds each record error type on the Diag report.
^TMP("PRPF_EXTDATA",$J,0)	Holds the purge date and the date the Extraction was last performed.
^TMP("PRPF_EXTDATA",$J,1,0)	Holds the single header record that is used by VistAMigrate
^TMP("PRPF_EXTDATA",$J,CNTSEG,CNTREC)	Holds each data record of the extraction file

4.4 Options

New options are brought in to support migration of the legacy Patient Funds data into VPFS.

Option	Description
Database Diagnostic Report [PRPF DATA DIAGNOSTIC REPORT]	Legacy Diagnostics menu option contained in Supervisors menu options.
[PRPF RPC UTILS]	Broker type option used to register VPFS Remote Procedure Calls used during the Data Migration process and control access to Diag and Extract RPC‟s.

4.5 Remote Procedure Calls

Name	Description
PRPF DATABASE DIAG	Broker type option used to register VPFS Remote Procedure Calls, which Data Diagnostic reports that are displayed in VistAMigrate.
[PRPF DATABASE EXTR]	Broker type option used to register VPFS Remote Procedure Call used during Data Migration

4.6 External Relations

KAAJEE: VPFS uses the Kernel Authentication Authorization Java Enterprise Environment, a Security service located on VistA for use by reengineered web applications, for authenticating users during sign on, and for allowing them access to various VPFS options based on roles. KAAJEE has a VistA component that must be installed in order to run VPFS. The roles-based access is controlled by the use of new security keys created for VPFS. See „Security Key‟ section below for details. The list of accessible divisions data is retrieved by KAAJEE from the DIVISION multiple on the VistA NEW PERSON file.

VistALink: VPFS uses VistALink, a communications bridge between VistA and J2EE application Servers, to retrieve data. VistALink has a VistA component that must be installed in order to run VPFS.

Person Services Lookup: VPFS uses Person Services Lookup service when the user needs to add a patient to VPFS. The service includes a User Interface (UI) that is incorporated within VPFS. The UI prompts for patient information, displays a list of matching patients, then when one is selected, displays any patient related warnings and additional information. The identifier for the patient is then returned to the VPFS application and stored in a local VPFS table. Person Services Lookup has a VistA component that must be installed in order to run VPFS.

Patient Services Construct: After a patient has been selected, VPFS calls Patient Services Construct service to get patient demographics information for the patient. This information is stored on a local VPFS table to support reporting. Patient Service Construct has a VistA component that must be installed in order to run VPFS.

Note: None of the new VistA routines, options or remote procedure calls brought in with the VPFS installation should be called by any application outside of VPFS.

4.7 Online Documentation

The VPFS options used for data migration use standard FileMan conventions for online help.

4.8 Checksum Values for Routines

Refer to the patch description for each patch for routine checksums.

4.9 Security and Keys

VPFS uses the KAAJEE security service to authenticate users during sign on and authorize access to the VPFS application based on their assigned roles. The same access and verify codes used for VistA are also used to sign on to VPFS.

Security Keys stored in the VistA system are mapped to roles in the application server to control access to various options within the VPFS system.

Keys	Access Level
PRPF_BASIC_OFFICIAL_USER	User has the ability to view selected patient and account information, no reporting privileges.
PRPF_BASIC_PFC	User has the ability to register patients, search for patients, edit patient information, post transactions, request patient transfers.
PRPF_LEAD_PFC	User has the ability to register patients, search for patients, edit patient information, post transactions, request and authorize patient transfers.
PRPF_PFC_SUPER	User has the ability to register patients, search for patients, edit patient information, post transactions, request and authorize patient transfers, request application changes through Administration area.
PRPF_FISCAL_MANAGEMENT	User has the ability to view selected patient and account information, no reporting privileges.
PRPF_VPFS_SECURITY_ADMIN	User has the ability to view selected patient and account information for purposes of data security.
PRPF_VPFS_SYSTEM_ADMIN	User has the ability to implement authorized changes to common reference data, no patient record access. **This is a restricted role.**
PRPF_ACCOUNT_OVERDRAW	User has the ability to overdraw any patient account.
PRPF_DEFERRAL_OVERRIDE	User has the ability to override deferred transactions.
PRPF_RESTRICTION_OVERRIDE	User has the ability to override patient restrictions.
PRPF_DATA_MIGRATION_USERS	User has the ability to migrate legacy Patient Funds data. **This is a restricted role.**

Database - Oracle

5 Database - Oracle

The VPFS persistent data is stored in Oracle tables maintained on a central server. The version at the time of VPFS initial installation is Oracle 12g.

5.1 Database

The VPFS production database is located at the Falling Waters EIE.

5.2 Schemas

VPFS – Owns all of the tables and procedures for the VPFS application. Also owns the data staging tables and conversion stored procedure needed for VistAMigrate.

VISTA_MIGRATE – Owns all of the tables and procedures for the VistAMigrate application.

5.3 Users

VPFS – Owns all of the tables and procedures for the VPFS application. All database connections from the VPFS application user interface are made as the VPFS user.

VISTA_MIGRATE – Owns all of the tables and procedures for the VistAMigrate application. All database connections from the VistAMigrate application user interface are made as the VISTA_MIGRATE user.

5.4 Roles

No roles are created for VPFS.

5.5 Tablespaces

No tablespaces are created for VPFS. (The VPFS schema can be installed into its own tablespace if one is created prior to starting installation and the VPFS user is assigned this tablespace as its default tablespace.)

5.6 Tables

5.6.1 Tables in the VPFS Schema

All of the data tables used in the VPFS application are in the VPFS schema.

Table Name	Description
ACCOUNT_TRANSACTION	Patient transaction records
FREQUENCY	System-wide list of income frequency types (daily, weekly, monthly, etc.)
GUARDIAN	Patient guardian records (VA and civil)
INCOME_SOURCE	Patient income source records
INCOME_SOURCE_TYPE	Institution-specific list of income source types (apportionee, guardian, institutional award, etc.)
PATIENT_ACCOUNT	Patient account records

Table Name	Description
PATIENT_EVENT_LOG	Patient event log records
PATIENT_FUNDS_FORM	Institution-specific list of form types
PATIENT_FUNDS_REMARK	Institution-specific list of remark types, when posting transactions
PATIENT_STATUS	System-wide list of patient status types (competent, etc.)
PATIENT_TRANSFER	Pending patient transfer details
PATIENT_TYPE	System-wide list of patient types (restricted, limited restricted, unrestricted, unknown)
PAYEE_TYPE	Institution-specific list of payee types
PAYMENT_TYPE	System-wide list of payment types (check/cash/other)
PERSON	Person identity records
REFERENCE_TYPE	Institution-specific list of transaction reference types (ACP, PFO, etc.)
STANDARD_INCOME_ SOURCE_TYPE	Standard list of income source types, available at every institution
STANDARD_PATIENT_ FUNDS_FORM	Standard list of form types, available at every institution
STANDARD_PATIENT_ FUNDS_REMARK	Standard list of remark types, when posting transactions, available at every institution
STANDARD_PAYEE_TYPE	Standard list of payee types, available at every institution
STANDARD_REFERENCE_TYPE	Standard list of reference types, available at every institution
SUSPENSE	Patient suspense item records
SYSTEM_ERROR_LOG	Error log available for use via log4j appender
USER_INSTITUTION	Cache for default child institution for application user
VPAGE	System table; list of individual application pages
VPAGE_FIELD	System table; page and field level help text
VPFS_INSTITUTION	System table; institution-specific data used by the application, including email address lists and messages for override notifications
VREPORT	System table; list of available reports, which roles can access them, and what parameters are available for them

VREPORT_PARAMETER	Deprecated
VSYSTEM_PARAMETER	System table; application configuration parameters

These tables are used by the VistAMigrate application for staging PFOP data prior to conversion. After data migration has been completed for all stations, these tables will no longer be needed.

Table Name	Description
DM_A1	Data migration header records
DM_B1	Patient balance records (1 of 2)
DM_B2	Patient balance records (2 of 2)
DM_D1	Patient detail / demographic records (1 of 2)
DM_D2	Patient detail / demographic records (2 of 2)
DM_I1	Patient income source records
DM_R1_R2	Patient general remarks
DM_S123	Patient suspense items
DM_T1	Patient deferred transactions being migrated
DM_X1_X2	Patient special remarks
DM_ERRORS	List of errors encountered during the Analyze step of migration

5.6.2 Tables in the VISTA_MIGRATE Schema

All of the data tables used in the VistAMigrate application are in the VistAMigrate schema.

Table Name	Description
DM_SETUP	Project parameters, RPC names, database connection information
DM_SETUP_INSTITUTION	Migration statistics and progress status per institution
VSYSTEM_PARAMETER	System table; application configuration parameters
DM_CONVERSION_EVENT_LOG	Event log for conversion step
DM_CONVERSION_ERROR_LOG	Error log for conversion step

5.7 Field Information

IDs: The primary data tables (person, patient_account, account_transaction, suspense, income_source, guardian) have a sequential integer ID field as the primary key. This field value is automatically incremented from an Oracle sequence for that table.

User IDs: Each table has a user_id field and last_modified_dt field to maintain the name of the user and date of the last update to the corresponding record.

5.8 Views

5.8.1 Views in the VPFS Schema

The following views used by the VPFS application are owned by the VPFS schema.

View Name	Description
WARD	Pick list of the distinct wards present in the patient accounts, by institution.
INSTITUTION	Consolidated view of institution data from the SDS STD_INSTITUTION table with VPFS-specific institution data from the VPFS_INSTITUTION table. The INSTITUTION view references the following SDS tables: • STD_INSTITUTION • STD_STATE • STD_COUNTRY • STD_FACILITYTYPE

5.9 Procedures

5.9.1 Stored Procedures in the VPFS Schema

The following procedures used by the VPFS application are owned by the VPFS schema.

Procedure Name	Description
sp_deactivateAccount	Checks if an account needs to be deactivated (if there is zero balance and there are no transactions for 30 days). Scheduled to run daily.
sp_instnOrderNbrOnAfterDelete	Reorder the sequence numbers of rows in an institution-specific maintenance table AFTER the deletion of a record.
sp_instnOrderNbrOnBeforeInsert	Reorder the sequence numbers of rows in an institution-specific maintenance table BEFORE the insert of a new record.
sp_instnOrderNbrOnBeforeUpdate	Reorder the sequence numbers of rows in an institution-specific maintenance table when updating a record.
sp_redoOrderNbrOnAfterDelete	Reorder the sequence numbers of rows in a maintenance table AFTER the deletion of a record.
sp_redoOrderNbrOnBeforeInsert	Reorder the sequence numbers of rows in a maintenance table BEFORE the insert of a new record.
sp_redoOrderNbrOnBeforeUpdate	Reorder the sequence numbers of rows in a

Procedure Name	Description
	maintenance table when updating a record.
sp_updateDeferrals	Checks all deferral dates and updates the proper balances. Scheduled to run daily.
sp_updateMonthlyRestriction	Update the actual_monthly_restriction_amt to 0. Scheduled to run monthly.
sp_updateWeeklyRestriction	Update the actual_weekly_restriction_amt to 0. Scheduled to run weekly.
logError	This procedure is called only by other stored procedures to log an error to the system error log table.

The following procedures are used for data migration (VistAMigrate) only.

Procedure Name	Description
pk_convertLegacyData	Package of procedures used to convert the data in the staging tables (DM_*) for the VPFS production tables. mainVPFS is called by VistAMigrate during the Conversion step. The remaining procedures (in *italics*) are called only by mainVPFS. **Contents**: mainVPFS *preConversionCheck* *prepDates insertPerson* *insertPatientAccount* *insertIncomeSource* *insertAccountTransaction* *insertSuspense* *updateRunningBalance* *updateDeferredAvailableBalance* *updateGeneralRemarks* *updateSpecialRemarks* *postConversionCheck* *logConversionError* *logConversionEvent* logConversionError and logConversionEvent call the logError and logEvent procedures, respectively, in the DM_UTILS package in the VISTA_MIGRATE schema. The VPFS user must have been granted execute privileges to the DM_UTILS package in the VISTA_MIGRATE schema.
sp_vpfs_cleanup_dm	Removes data for the specified station_id from the

Procedure Name	Description
	staging tables.
sp_vpfs_cleanup_production	Removes data for the specified station_id and the children of the specified station_id from the production tables.
sp_vpfs_converted_count	Returns the number of converted patient accounts and the total number of converted patient records (including transactions, income sources, etc.).
sp_vpfs_stage_count	Returns the number of staged patient accounts and the total number of staged patient records (including transactions, income sources, etc.).

5.9.2 Stored Procedures in the VISTA_MIGRATE Schema

The following procedures used by the VistAMigrate application are owned by the VISTA_MIGRATE schema.

Procedure Name	Description
pk_DM_UTILS	Package of procedures used by VistAMigrate to update various record counts and manage migration event and error logs. **Contents**: dmSetup updateExtractionCount updateLoadCount updateMoveCount cleanupLogs logError logEvent

5.10 Triggers

5.10.1 Triggers in the VPFS Schema

The following triggers are defined in the VPFS schema. These triggers provide pre- and postprocessing for deletes, inserts, or updates on data in the VPFS tables.

Trigger Name	Description
del_standard_income_src_type	• Fired when a row is deleted from the standard_income_source_type table. • Updates the standard_data_ind flag on rows in the income_source_type table having the same income_source_type_cd as the row being deleted from the standard_income_source_type table. • Allows individual stations to choose to include or

Trigger Name	Description
	not include the deleted standard income source type.
del_standard_patient_funds_frm	• Fired when a row is deleted from the standard_patient_funds_form table. • Updates the standard_data_ind flag on rows in the patient_fund_form table having the same form_id as the row being deleted from the standard_patient_funds_form table. • Allows individual stations to choose to include or not include the deleted standard patient funds form type.
del_standard_patient_funds_rmk	• Fired when a row is deleted from the standard_patient_funds_remark table. • Updates the standard_data_ind flag on rows in the patient_funds_remark table having the same rmks_cd as the row being deleted from the standard_patient_funds_remark table. • Allows individual stations to choose to include or not include the deleted standard remark type.
del_standard_payee_type	• Fired when a row is deleted from the standard_payee_type table. • Updates the standard_data_ind flag on rows in the payee_type table having the same payee_type_cd as the row being deleted from the standard_payee_type table. • Allows individual stations to choose to include or not include the deleted standard payee type.
del_standard_reference_type	• Fired when a row is deleted from the standard_reference_type table. • Updates the standard_data_ind flag on rows in the reference_type table having the same reference_txt as the row being deleted from the standard_reference_type table. • Allows individual stations to choose to include or not include the deleted standard reference type.
ins_acct_amt	• Fired when a row is inserted into the account_transaction table. • Updates the total balance, available balance, deferred balance, private source and gratuitous balances, and actual monthly and weekly restriction amounts of the patient_account associated with the transaction.

Trigger Name	Description
	• Updates the running balance in the transaction • Sets the account status of the patient account to active („A‟).
ins_acct_trans	• Fired when a row is inserted into the account_transaction table. • Updates the default institution for transaction for the current user.
ins_standard_income_src_type	• Fired when a row is inserted into the standard_income_source_type table. • Adds the new standard income source type to the income source type list for each institution *or* • Sets the standard_data_ind flag to „Y‟ if the income_source_type_cd already exists at the institution.
ins_standard_patient_funds_frm	• Fired when a row is inserted into the standard_patient_funds_form table. • Adds the new standard form type to the form type list for each institution *or* • Sets the standard_data_ind flag to „Y‟ if the form_id already exists at the institution.
ins_standard_patient_funds_rmk	• Fired when a row is inserted into the standard_patient_funds_remark table. • Adds the new standard remark type to the remark type list for each institution *or* • Sets the standard_data_ind flag to „Y‟ if the rmks_cd already exists at the institution.
ins_standard_payee_type	• Fired when a row is inserted into the standard_payee_type table. • Adds the new standard payee type to the payee type list for each institution *or* • Sets the standard_data_ind flag to „Y‟ if the payee_type_cd already exists at the institution.
ins_standard_reference_type	• Fired when a row is inserted into the standard_reference_type table. • Adds the new standard reference type to the reference type list for each institution *or* • Sets the standard_data_ind flag to „Y‟ if the reference_txt already exists at the institution.
ins_suspense	• Fired when a row is inserted into the suspense table. • Updates the max suspense date

Trigger Name	Description
	(max_suspense_dt) for the patient account associated with the suspense item.
ins_upd_acct_trans	• Fired when a row is inserted or updated in the account_transaction table. • Updates the last transaction date (last_trans_entered_dt) and last activity date (last_activity_dt) for the patient account associated with the transaction.
ins_upd_pat_acct	• Fired when a row is inserted or updated in the patient_account table. • Updates the patient account station_cd to ensure consistency with the current instn_id by crossreferencing the institution view (which references the SDS std_institution table).
upd_deferral_dt	• Fired when the deferral_dt is changed in an account_transaction record. • Updates the available and deferred balances for the patient account associated with the transaction based on the old and new deferral dates.
upd_director_payee_ind	• Fired when the director_payee_ind is changed in a guardian record. • Updates the (internal) director_payee_exists_ind indicator in patient_account.
upd_patient_instn	• Fired when the instn_id is changed in a patient_account record. • Adds an entry to the patient event log when the patient is assigned to a different institution.
upd_patient_status	• Fired when the patient_status_cd is changed in a patient_account record. • Adds an entry to the patient event log when the patient status is changed.
upd_patient_type	• Fired when the patient_type_cd is changed in a patient_account record. • Adds an entry to the patient event log when the patient type is changed.
upd_standard_income_src_type	• Fired when a row is updated in the standard_income_source_type table. • Updates the data for rows in the patient_funds_form table having the same income_source_type_cd as the row being updated.

Trigger Name	Description
upd_standard_patient_funds_frm	• Fired when a row is updated in the standard_patient_funds_form table. • Updates the data for rows in the patient_funds_form table having the same form_id as the row being updated.
upd_standard_patient_funds_rmk	• Fired when a row is updated in the standard_patient_funds_remark table. • Updates the data for rows in the patient_funds_remark table having the same rmks_cd as the row being updated.
upd_standard_payee_type	• Fired when a row is updated in the standard_payee_type table. • Updates the data for rows in the patient_funds_remark table having the same payee_type_cd as the row being updated.
upd_standard_reference_type	• Fired when a row is updated in the standard_reference_type table. • Updates the data for rows in the patient_funds_remark table having the same reference_txt as the row being updated.
upd_ward_nm	• Fired when the ward_nm is changed in a patient_account record. • Adds an entry to the patient event log when the patient ward is changed.

5.11 Scheduled Jobs

5.11.1 Jobs in the VPFS Schema

The following scheduled jobs are defined in the VPFS schema.

Job Action	Description	Schedule
sp_deactivateAccount	Checks if an account needs to be deactivated (if there is zero balance and there are no transactions for 30 days).	Daily, at 2:00 AM
sp_updateDeferrals	Checks all deferral dates and updates the proper balances.	Daily, at 2:00 AM
sp_updateMonthlyRestriction	Update the patient_account actual_monthly_restriction_amt to 0.	Monthly, on the 1st of the month
sp_updateWeeklyRestriction	Update the patient_account actual_weekly_restriction_amt to 0.	Weekly, Sunday morning at midnight

5.12 External Dependencies

VPFS requires that the following dependencies be installed in the VPFS Oracle database.

5.12.1 Standard Data Service (SDS)

VPFS has subscribed to the SDS group to load their standard data tables into the SDSADM schema on the VPFS Oracle database. The data will be automatically refreshed by the SDS application. VPFS uses supported SDS APIs to extract data from the tables as needed.

The INSTITUTION view in the VPFS schema references the std_institution, std_state, std_country and std_facilitytype tables. This view is used throughout the application.

5.13 Locking

The VPFS application uses optimistic locking, meaning for most cases, “last in wins”. If, for example, you have two users accessing the same patient (“A”), and they both attempt to update the same value (“Type”) at the same time, the updates will be processed in the order received, so the user whose update is processed *last* will see their value as the new value.

	(1) Page displayed: Type is “U”	**(2)** User 1 changes Type to “L”	**(3)** User 2 changes Type to “R”	**(4)** Page redisplayed: Type is “R”
User 1	“U”	“L”	“L”	“R”
User 2	“U”	“U”	“R”	“R”

Time →

In this case, User 2‟s update was processed last, so when the page is redisplayed, both users will see User 2‟s update.

When posting a patient funds transaction, however, the database transaction must be controlled more carefully. To make sure the account cannot be overdrawn if two users were to attempt to post transactions for the same patient at the same time (without requiring authorization from the user), the account balance is checked just prior to inserting the transaction in the database, using same database transaction. This ensures that the balance has not changed between the time the balance was initially read and when the balance is updated after posting the transaction.

	(1) Page displayed: Balance $100	**(2)** User 1 withdraws $75	**(3)** User 2 withdraws $50	**(4)** Page redisplayed: Balance $25
User 1	$100	$25	$25	$25
User 2	$100	$100	Error: Overdrawal	$25

Time →

In this case, User 2 sees an overdrawal error (step 3) because when the balance is re-read prior to posting the transaction, User 1‟s transaction has already posted (step 2), reducing the actual balance to $25, therefore there are no longer sufficient funds to process User 2‟s transaction

6 Integration Agreements

Integration Agreements for Health*e*Vet dependency services are still being developed. Those that have been made available are listed in this section.

6.1 Supported

4851 – KAAJEE – VPFS consumes the KAAJEE service to manage authentication and authorization controls. Interaction with this service occurs via documented objects and APIs.

4955 – Electronic Signature – VPFS consumes the Electronic Signature (ESig) service to handle Electronic Signature verification and updating. Interaction with this service occurs via documented objects and APIs.

6.2 Controlled Subscription

Not applicable at this time.

6.3 Private

4949 – Standard Data Service – VPFS consumes the Standard Data Service (SDS) to retrieve institution data. Interaction with this service occurs via documented objects and APIs. This agreement also documents the use of a read-only database view in the VPFS schema that incorporates the following tables owned by SDS: std_institution, std_facilitytype, std_state, and std_country.

4951 – Person Service Lookup – VPFS consumes the Person Service Lookup (PSL) service to manage patient and provider lookups from VistA. Interaction with this service occurs via documented objects and APIs.

4952 – Person Service Construct – VPFS consumes the Person Service Construct (PSC) service to retrieve patient demographic data from VistA. Interaction with this service occurs via documented objects and APIs.

4974 – VistALink (Java API) – VPFS consumes the VistALink service as required by the Person Service Lookup, Person Service Construct, Electronic Signature, and KAAJEE services. VistAMigrate consumes the VistALink service to execute the M Diagnostic and M Extract RPCs as part of the migration process. Interaction with this service occurs via documented objects and APIs.

7 Patient Demographics Updates

VPFS stores some patient demographic data to facilitate user and reporting needs. However, VPFS does not “own” this data and is not the authoritative source of patient demographic data. Therefore, a couple processes were developed to ensure the patient demographic data stored remains current by retrieving that data from the owning VistA site.

7.1 Data Updated

The following patient demographic data is updated by the patient update process:

- Identity – name, SSN, ICN, DFN/IEN, date of birth
- Address* – main and temporary patient addresses
- Primary Demographics – gender, date of death
- ADT – admission and discharge dates, ward, room/bed
- Eligibility Information – service connected percent, claim number, person type, veteran status
- Incompetence Information* – date ruled incompetent, VA and civil guardian name and addresses

This data is in the PERSON, PATIENT_ACCOUNT, and GUARDIAN tables. Refer to Appendix A for more detail.

* This information is only updated on-demand. It is not updated during the scheduled update.

7.2 On-demand Update

Whenever a patient is selected in the VPFS application from the Select Patient page, or by using the Next/Previous buttons available from any Patient tab page, the patient demographic data for the selected patient, including address and incompetence information, is updated from the owning VistA site using the Patient Service (PSC).

The patient update process starts by getting the ICN of the selected patient. The PSC retrievePatientData method is used to retrieve the patient data for the patient with the specified ICN. If a patient with the specified ICN could not be found, the lookup is repeated using the patient DFN. If the patient still could not be found, an error results, otherwise the patient data is updated in the VPFS database.

7.3 Scheduled Update

A Scheduled Patient Update process is started when the application is started on the application server. This process handles updating the patient demographic data for patients registered in VPFS. Address and incompetence information is not updated during the scheduled patient update process. This information is updated on-demand.

Patient demographics for *active* patients are updated daily.

Patient demographics for *inactive, non-deceased* patients are updated weekly (over the weekend).

Patient demographics for *inactive, deceased* patients are only updated on-demand when their account is selected by a user in VPFS.

The following parameters are used to configure the update process. These parameters can be modified via the Maintain System Parameters page in VPFS.

Parameter Name	Description	Default Value
PATIENTUPDATE_SCHEDULE_TIME	Start time of the scheduled update process.	1:00 am
APPLICATION_PROXY_USER	VistALink application proxy user account, required for PSC to be able to connect to the VistA sites.	VPFS,APPLICATION PROXY

Note: The application proxy user name must exist on every VistA site VPFS will connect to in order to update patient demographic data at that site.

The scheduled patient update process is a batch version of the on-demand update that runs in a separate timer thread that is scheduled when VPFS is started in the application server. This process can take a significant amount of time due to the amount of data being processed.

The process starts by retrieving a list of station numbers of patients present in the VPFS database. For each station, the ICNs of patients at that station are retrieved. These are then “chunked” into 1000 patient batches and passed to the PSC retrieveMultiplePatients method. Any patient not found via ICN lookup will be processed singly (using the on-demand update process) using their DFN. An error message will be written to the VPFS log file for any patient that could not be found or updated.

Because the scheduled patient update process is executed outside of an active user session, an application proxy user account is used. The same application proxy user account name must be present in the VistA database for every station. This account is created automatically during the installation of PRPF*3.0*16.

8 Data Migration

The VPFS Data Migration process is documented in the *VistAMigrate Data Migration Guide*.

8.1 Configuration

VistAMigrate configuration is documented in the *VPFS Installation Guide*.

The following parameters are configurable for the VPFS project in VistAMigrate.

Location	Parameter Name	Description
DM_SETUP	DB_USER_NM	VPFS database username (user with read/write access to the VPFS schema, e.g. `VPFS`)
DM_SETUP	DB_USER_PASS_TXT	VPFS database password (password for the user specified in `DB_USER_NM`)
DM_SETUP	DB_DATA_SRC_TXT	TNS name defined on the application server for the VPFS database (user/password@**tnsname** when using SQL*Plus).
VistAMigrateResources.properties (in WEB-INF/classes)	migrationRootPath	Path to the VistAMigrate root path (directory containing the `VPFS/CODE` directory, e.g. `/path/to/vistamigrateroot`).
setenv (in VistAMigrate root VPFS/CODE directory)	VM_ORACLE_HOME	Path to the Oracle installation root path (directory containing the `bin` directory, e.g. `/path/to/oracle/product/10.2.0/client_1`).

9 VPFS Troubleshooting

Component	Message	Cause	Resolution
KAAJEE	**Refer to the KAAJEE Deployment Guide for additional information about these KAAJEE related errors or for a loginrelated error not listed here.**		

KAAJEE	**Error: Not a valid ACCESS CODE/VERIFY CODE pair**	The user has entered an incorrect access and/or verify code or does not have access to the selected VistA M Server.	The user should re-enter their access and verify codes. If the error persists, the user *must* contact IRM or the System Administrator to verify that they are allowed access to the VistA M Server account in question and have been granted appropriate access.
KAAJEE	**Error: Your verify code has expired or needs changing**	The user's verify code has expired or the user has been given a temporary verify code because they are new to the VistA M Server or asked IRM to give them new access. Upon their first login, this temporary Verify code expires immediately and *must* be changed.	KAAJEE-enabled Web-based applications do *not* support changing the user‟s verify code at this time, users *must* use another *non*-KAAJEE-based application to change their Verify code.
KAAJEE	**Error: Institution/division you selected for login is not valid for your M user account**	The user does not have the selected Institution/Division entry in the DIVISION Multiple field (#16) in the NEW PERSON file (#200) entry.	The user *must* contact IRM or the System Administrator to verify that they are allowed access to the Institution/Division in question and have been granted appropriate access.

Component	Message	Cause	Resolution
KAAJEE	**Error: Forms authentication login failed**	The user's account does not have the necessary VistA M Server J2EE Security Keys required to access the requested page.	Get the necessary VistA M Server J2EE Security Keys assigned. **VistAMigrate Keys:** PRPF_DATA_MIGRATION_USER **VPFS Keys:** PRPF_BASIC_OFFICIAL_USER PRPF_BASIC_PFC PRPF_LEAD_PFC PRPF_PFC_SUPER PRPF_FISCAL_MANAGEMENT PRPF_VPFS_SYSTEM_ADMIN PRPF_VPFS_SECURITY_ADMIN
KAAJEE	**Error: Login failed due to too many invalid logon attempts**	The user has exceeded the allowed number of login attempts to the VistA M Server and *must* wait a prescribed period of time before attempting another login.	If after the prescribed wait period has passed and the user tries to log back into the VistA M Server, and again fails in the attempt, the user *must* contact IRM or the System Administrator for assistance.
KAAJEE	**Error: Logins are disabled on the M system**	IRM or the System Administrator has disabled logins on the VistA M Server. Logins are sometimes disabled in order to install new software or perform system maintenance.	The user should wait and try to log into the VistA M Server at a later time. If the user feels the time period to log back into the system is excessive, the user should contact IRM or the System Administrator for assistance.

Component	Message	Cause	Resolution
KAAJEE	**Error: Could not get a connection from connector pool**	No VistALink connector is configured for the Station Number selected or the VistA M Server to which the connector is connecting is down.	Contact the Systems Administrator for assistance. Check that a VistALink connector is deployed for the specified Station Number and that the VistA M Server is available.
VPFS			
VPFS	**System Error**	An unexpected error occurred. The most common causes for this error include: • System configuration problem. • VPFS Oracle database is unavailable. • VistA M Server is unavailable.	Contact the Systems Administrator for assistance. Check the application log file for error details. Refer to the Logging section of this document for additional information about how errors are logged.
VPFS	**The application encountered a general VPFS Exception.**	An unexpected error occurred. These errors are truly exceptional conditions that generally point to a defect in the application.	Contact the Systems Administrator for assistance. Check the application log file for error details. Refer to the Logging section of this document for additional information about how errors are logged.
	The application encountered a Business Delegate Code Exception.		
	The application encountered a VPFS System Exception.		
	The application encountered a Security Exception.		

	The application encountered a JSP Tag Exception.		
	The application encountered a Service Locator Exception.		

Component	Message	Cause	Resolution
	The application encountered a Database Exception.	Possible causes include: • VPFS Oracle database is unavailable • Key violation (duplicate, foreign)	Contact the Systems Administrator for assistance. Check the application log file for error details.
VPFS	**Foreign Key Exception**	Foreign Key Exception when starting VPFS after logging in.	The station number the user is attempting to log into exists in the SDS institution table but does not exist in the vpfs_institution table.
VPFS	**The demographics information for this patient could not be updated. {x} Please notify the system administrator of this problem.**	Possible causes include: • Patient SSN has changed • Patient IEN has changed • Patient Name has been set to blank.	To maintain integrity of the patient data, the patient SSN and IEN cannot be changed as part of the demographic update. If the SSN or IEN need to be changed, contact the DBA for assistance.
VPFS	**There is a discrepancy between the computed and stored balances for patient {x}.**	The patient account balance does not reconcile with the transaction data for that patient. This is a major problem that indicates the stored balance(s) on the patient account may have been changed or one or more transactions may have been deleted.	Review the patient account. The cause for the discrepancy must be accounted for and the balance or transactions must be corrected. Any changes to the patient account balances would need to be made by the DBA and logged for auditing purposes.

VPFS	**There is a problem verifying the integrity of the transactions for patient {x}.**	The transaction checksum could not be verified; transaction data has been modified. This is a major problem that indicates the transaction data may have been changed.	Review the patient account. The cause for the discrepancy must be accounted for. Any changes to the patient account or transactions would need to be made by the DBA and logged for auditing purposes.

Component	Message	Cause	Resolution
VPFS	**The application encounted a problem sending email. Please contact your System Administrator. {x}**	This exception can occur when posting transactions that have caused an overdrawal, or overridden a restriction or deferral. Possible causes include: • The SMTP server has not been configured correctly or is unavailable. • No recipients have been listed for the current institution.	Have the VPFS System Administrator check the SMTP_SERVER property in the Maintain System Parameters page. Ensure the server is correct and available. Check the list of recipients for the current institution in the Maintain Institutions page. If no recipients have been specified, determine who should receive override messages and add their complete email address to this list.
VPFS	**A patient with this ICN or SSN is already registered in VPFS at station {x}.**	VPFS does not allow a patient to be registered at multiple stations. VA policy states that a patient may have only one patient funds account. However, PFOP, which is installed at each site, has no way of knowing if a patient has a patient funds account at another site. VPFS, being a centralized application, can determine what the "owning station" is for the patient funds account.	If the patient is currently registered in VPFS and has been assigned to a station that is not the "owning station", contact the PFC at the station the patient is currently assigned to. That PFC can change the station the patient is assigned to via the Station list on the Patient tab.

VPFS	**This patient has a pending transfer request by {x} at station {x}.**	The user requested the transfer of a patient that has a pending transfer request.	There can only be one pending transfer request for a patient at a time. If the user would like to transfer a patient that has a pending transfer request, they need to wait for that transfer to be completed or cancelled. They can contact the PFC at the station responsible for completing the outstanding request.

Component	Message	Cause	Resolution
VPFS	**You have not been authorized to access this patient. Contact the Patient Funds department at station {x}.**	The user clicked the View Patient button on the Request Transfer page, however the transfer has not yet been authorized by the station the patient is currently assigned to.	Contact the PFC at the station that the patient is being requested to transfer *from* to have them authorize the transfer. Once they have authorized the transfer, the requestor will be able to click the View Patient button.

10 VistAMigrate Troubleshooting

Component	Message	Cause	Resolution
KAAJEE	**Possible KAAJEE errors are the same as in VPFS.**		
VistAMigrate			
VistAMigrate	**The M Diagnostic or M Extract step fails, the step number is in a red circle.**	The migration root path is not set correctly or the WebLogic server does not have write permissions to that directory. The user account for the current user has not been completely set up and may not have the appropriate security keys or menu options to run VistAMigrate.	Review the VPFS Installation Guide to ensure VistAMigrate has been completely set up. Ensure the WebLogic server migration root path has write privileges. Check that the user account has the PRPF RPC UTIL secondary menu option.
VistAMigrate	**The Stage Data step fails, the step number is in a red circle.**	The migration root path is not set correctly The database username, password, or data source (TNS) name are not set correctly The Oracle home path is not set correctly. The weblogic user does not have permissions to execute the Oracle client tools (sqlldr, sqlplus).	Ensure that all paths and database connection information are set correctly and the weblogic user has execute permissions on the Oracle client tools.
VistAMigrate	**The Analyze Data step fails, the step number is in a red circle.**		

Component	Message	Cause	Resolution
VistAMigrate	**The Convert Data step fails, the step number is in a red circle.**	Check the number of D1 records, the number of B1 records, and the number of patients in the A1 (header) record are not equal	Check the Stage Data log. If any D1 records were dropped, the data that caused the record to be dropped must be corrected. Typical causes for a D1 record to be dropped include control characters (carriage return or line feed) or field value too long. Also check for dropped B1 records, however this condition should be caught in the Analyze Data step and cause the patient not to be migrated.

Appendix A Patient Data Field Mapping

The following table maps where patient data retrieved from VistA via PSC is updated in the VPFS database.

	PSC Field(s)			VPFS Database Fields
Identity Information	full name			person.full_nm, patient_account.patient_full_nm, patient_account.patient_upper_last_nm patient_account.patient_cd (partial)
	DOB			patient_account.birth_dt
	IEN / DFN			person.ien_id patient_account.ien_id
	ICN			person.icn patient_account.icn
	SSN			person.ssn_nbr patient_account.patient_cd (partial)
Address Demographics	*Main Address*			patient_account: addrs_one_txt, addrs_two_txt, addrs_three_txt, city_nm, postal_cd, zip_prefix_nbr, home_phone_nbr, work_phone_nbr
	addressOne addressTwo addressThree	city state zip	homePhone workPhone	
	VA Guardian Address			guardian: addrs_one_txt, addrs_two_txt, addrs_three_txt, city_nm, postal_cd, zip_prefix_nbr, home_phone_nbr, work_phone_nbr
	addressOne addressTwo addressThree	city state zip	homePhone workPhone	
	Civil Guardian Address			guardian:

	addressOne addressTwo addressThree	city state zip	homePhone workPhone	addrs_one_txt, addrs_two_txt, addrs_three_txt, city_nm, postal_cd, zip_prefix_nbr, home_phone_nbr, work_phone_nbr
	Temporary Patient Address			patient_account:

	PSC Field(s)			VPFS Database Fields
	addressOne addressTwo addressThree	city state zip	homePhone workPhone	temp_addrs_one_txt, temp_addrs_two_txt, temp_addrs_three_txt, temp_city_nm, temp_postal_cd, temp_zip_prefix_nbr, temp_home_phone_nbr, temp_work_phone_nbr
Primary Demographics	gender			patient_account.gender_cd
	DOD			patient_account.death_dt
ADT	admission date			patient_account.admission_dt
	discharge date			patient_account.discharge_dt
	room / bed			N/A
	ward			patient_account.ward_nm
Enrollment Eligibility	service connected %			patient_account.service_connected_percent_nbr
	primary eligibility code			patient_account.primary_eligibility_txt
	claim#			person.claim_nbr
	person type code			person.person_type_cd patient_account.person_type_cd
	Veteran status			patient_account.veteran_status_ind
Incompetence Information	civil date ruled incompetent			guardian.rule_incompetent_dt
	civil guardian name			guardian.full_nm
	civil guardian relationship			guardian.rlnshp_nm
	civil guardian address			guardian (see Civil Guardian Address, above)
	VA guardian name			guardian.full_nm
	VA guardian relationship			guardian.rlnshp_nm

	PSC Field(s)	VPFS Database Fields
	VA guardian address	guardian (see VA Guardian Address, above)

Appendix B VPFS Security Roles Matrix

The following table maps available VPFS functions by role. A user account may have access to one or more roles, as determined by the security keys assigned to that user account in VistA. For more information about the security keys for VPFS, see the VPFS User Guide.

Menu Item	Sub Item 1	Sub Item 2	Basic Official Users	Basic PFC	Lead PFC	PFC Super	Fiscal Management	VPFS System Admin.	VPFS Security Admin.	*Overdraw Account*	*Override Deferral*	*Override Restriction*
Select Patient												
	Search and Filter		X	X	X	X	X		X			
	View All Patients (per Institution)		X	X	X	X	X		X			
	Select Patient, go to Account page for patient record		X	X	X	X	X		X			
	Show Patient Cards		X	X	X	X						
	Register Patient			X	X	X						
	Request Patient Transfer			X	X	X						
	Post Multiple Transactions			X	X	X						
		Overdraw Account								+		
		Override Restriction										+
		Override Deferral Date									+	

	Verify Balances				X	X	X					
Account Tab												
	View Account Balance		X	X	X	X	X		X			
	Post Transaction			X	X	X						
		Overdraw Account								+		
		Override Restriction										+

Menu Item	Sub Item 1	Sub Item 2	Basic Official Users	Basic PFC	Lead PFC	PFC Super	Fiscal Management	VPFS System Admin.	VPFS Security Admin.	*Overdraw Account*	*Override Deferral*	*Override Restriction*
		Override Deferral Date									+	
	Verify Balance				X	X	X					
	Transfer Patient				X	X						
Patient Tab												
	Edit Patient Information			X	X	X						
		Edit Special Remarks		X	X	X						
		Edit Station (Transfer Patient)			X	X						
	View Patient Information		X	X	X	X	X		X			
		View Special Remarks		X	X	X						
Guardian Tab												
	Add/Edit Guardian Information			X	X	X						
	View Guardian Information		X	X	X	X	X		X			
Transactions Tab												

Menu Item	Sub Item 1	Sub Item 2	Basic Official Users	Basic PFC	Lead PFC	PFC Super	Fiscal Management	VPFS System Admin.	VPFS Security Admin.	*Overdraw Account*	*Override Deferral*	*Override Restriction*
	Search and Filter			X	X	X	X		X			
	View Transaction History			X	X	X	X		X			
	View Transaction Reports			X	X	X	X					
	Edit Deferral Date				X	X						
Suspense Tab												
	Add Item			X	X	X						
	Cancel Item			X	X	X						
	Cancel Recurrences of Item			X	X	X						
	Cancel Items for Date			X	X	X						
	Review Suspense Items per Patient (Select Suspense)			X	X	X						
Log Tab												
	View Patient Log			X	X	X			X			
Administration												

Menu Item	Sub Item 1	Sub Item 2	Basic Official Users	Basic PFC	Lead PFC	PFC Super	Fiscal Management	VPFS System Admin.	VPFS Security Admin.	*Overdraw Account*	*Override Deferral*	*Override Restriction*
	Perform Table Maintenance											
		Award Frequencies						X				
		Form Types				X		X				
		Help Text						X				
		Income Source Types				X		X				
		Institutions				X		X				
		Patient Statuses						X				
		Patient Types						X				
		Payee Types				X		X				
		Payment Types						X				
		Remarks				X		X				
		References				X		X				
		System Parameters						X				
	User Account / Elect. Signature			X	X	X						

Reports												
	Patient Card (Selected Patient)											
		Print Selected Cards	X	X	X	X						
		Print All Cards	X	X	X	X						
		Transaction Display	X	X	X	X						
		Information Display	X	X	X	X						
		Master Transaction Review	X	X	X	X						
	Standard Reports											
		Activity Audit Listing		X	X	X	X					
		Dormant Account Listing		X	X	X	X					
		Indigent Patient Listing		X	X	X	X					
		Overdue Restriction Search		X	X	X	X					
		Patient Summary		X	X	X	X					
		Search for Min/Max Restrictions		X	X	X	X					
		Patient Listing		X	X	X	X					
		Account Balances		X	X	X	X					
		Transaction Listing		X	X	X	X					
		Fiscal Audit		X	X	X	X					

Menu Item	Sub Item 1	Sub Item 2	Basic Official Users	Basic PFC	Lead PFC	PFC Super	Fiscal Management	VPFS System Admin.	VPFS Security Admin.	*Overdraw Account*	*Override Deferral*	*Override Restriction*
		Fiscal Transaction Summary		X	X	X	X					
		Date Variance		X	X	X	X					
		Productivity				X	X					
		Negative Balances				X	X					
		Deceased Patient w/ Balance		X	X	X	X					
		Discharged Patient w/ Balance		X	X	X	X					
		Inactive Withdrawal Listing		X	X	X	X					
		Station Balances		X	X	X	X					
Select Suspense												
	Search and Filter			X	X	X	X					
	View Suspense Records across Patients			X	X	X	X					
	Select Suspense Item, go to Suspense page for patient record			X	X	X	X					

	Cancel Suspense Item			X	X	X						
Logout			X	X	X	X	X	X	X			