Enterprise Health Management Platform (eHMP)

Installation Guide for Release 1.2

Department of Veterans Affairs

May 2015

Version 2.2

Revision History

Date	Version	Description	Author
05/26/2015	2.2	Updated for Release 1.2 and submitted	ASM Research
03/24/2015	2.1	Submitted for 1002AF, 1003AF, 2004AF, 1004AF, 1006AF, 1008AF, 1009AF	ASM Research
12/01/2014	2.0	Submitted for 1004PMAS, 1006PMAS, 1008PMAS, 1009PMAS	ASM Research

Table of Contents

Table of Figures

Table of Tables

1. Introduction

The Enterprise Health Management Platform (eHMP) project is a multi-year effort to evolve a modern, service-oriented platform which provides a web-based user interface, clinical data services, and assembles patient clinical data from federated Veterans Health Information Systems and Technology (VistA) repositories and Department of Defense (DoD) data sources, reflective of each location providing care to the patient. This federated data is aggregated into an Enterprise Virtual Patient Record (eVPR). By leveraging and integrating elements of the Joint Legacy Viewer (JLV) and Health Management Platform (HMP), eHMP service components will span all application layers, including presentation, business and core services, and data access.

eHMP version 1.2 introduces a new architecture for managing the synchronization of patient and operational data. This architecture is based on Node.js, and replaces the previous Java-based architecture. Additional, significant changes for release 1.2 are the consolidation of mock services into a single virtual machine (VM), and the introduction of a central auditing repository server. Note that mock services are only relevant to a testing deployment of the application in which external connections to VistA, JMeadows, HDR/CDS, MVI and other systems are not available or desired.

1.1. Purpose

This Installation Guide outlines and details the procedures involved in installing the eHMP system into a VA-hosted environment that has been pre-configured with the required hardware and software. It also describes the process of installing the eHMP-specific VistA artifacts into a VistA site. This document is a step-by-step guide for use by the personnel responsible for the installation to perform a full deployment to bring eHMP to an operational state.

1.2. Scope

The Installation Guide provides the instructions for setting up a deployment environment required to:

- complete the eHMP deployment
- obtain access to the required VMs, provided in a VA environment such as the Austin Information Technology Center's (AITC) EO environment
- link Vagrant to the VMs
- provision the VMs

This guide also provides:

- the prerequisites required for the installation
- preparations required for the deployment
- instructions for deploying and provisioning the eHMP VMs
- instructions for deploying VistA M routines and patches

1.3. Assumptions

This guide assumes that the personnel responsible for the installation possess the following:

- A deployment workspace with connectivity to the VMs being deployed
- A stable build of eHMP
- Source Management Code (SMC) access to the VA's network
- Appropriately configured VMs
- The IP addresses of the VMs
- User accounts with Sudo privileges on each VM
- Access to the VistA KIDS builds and patches required for eHMP.

These assumptions are detailed in Section 2.

2. Prerequisites

This section details each of the requirements listed in Section 1.3. Each prerequisite should be met before beginning the installation of eHMP.

2.1. Deployment Workspace

Performing the deployment requires a deployment workspace on which the workspace setup script has been executed, 80 GB of disk space, and a user account within that environment that has Sudo access and Git cloning capabilities. The workspace setup script configures the server with the proper pre-deployment dependencies required to conduct the deployment. Eighty GB of disk space is required to hold the deployment artifacts. Sudo gives the user the security privileges of a super user or root user. Git cloning capabilities are required to pull in pre-deployment dependencies. When deploying to the Enterprise Development Environment (EDE) and to EO, the local laptop workspace is used.

2.2. Stable Build of eHMP

When performing the eHMP installation, the administrator will choose a specific version (build) to deploy. The administrator must select a stable build in order to avoid compatibility and integration issues.

2.3. System Management Controller (SMC) Access

VA SMC access with enhanced capabilities is required to perform the installation. CAG access is not sufficient to complete the deployment of the system.

2.4. Virtual Machines

Proper installation requires an appropriately configured VM for each server within the VA. It is important that each VM be configured with sufficient disk space for each required file system. At this time, it is assumed that the VMs will be set up and correctly configured by the VA within the EDE and EO environment.

2.5. IP Addresses

The administrator performing the installation will need the IP addresses of each VM to be deployed.

2.6. User Account with Sudo Privileges

Deployment of the VMs requires a local user account on each VM with the ability to use substitute user (sudo) commands. Sudo gives the user the security privileges of a super or root user. Sudo must be configured on the VMs such that executing sudo commands does not require the input of a password.

2.7. Access to VistA KIDS Builds and Patches

These eHMP-specific VistA artifacts are uploaded to a central FTP site where they can then be accessed and installed into any VistA site. Contact the eHMP sustainment team to request the URL and access to this FTP site.

3. Data Center Installation

The following sections describe the process for deploying the eHMP application into a VA data center such as AITC, PITC, etc. These components are the non-VistA parts of the application. A separate section below will discuss the installation of VistA M routines and patches into a VistA site.

3.1. Deployment Preparations

The following VM and user account preparations must be performed before deploying to any environment for the first time. Typically, these preparations are performed once, and need not be repeated.

3.1.1. Prepare Target Virtual Machines

Changes must be made to the server configuration before deployment. The following steps must be performed on each VM in the EDE and EO environments (if they have not been performed previously).

3.1.2. Select Virtual Machines to Deploy

Different servers may be deployed within different environments, depending on the unique requirements and objectives of that environment. Before deploying, have an understanding of which servers are to be deployed to which VMs.

The following servers are part of the core of eHMP for release 1.2, and should always be deployed:

- JavaScript Object Notation Data Store (JSON JDS)
- Searching On Lucene w/Replication (Solr)
- Vx-sync Server (Node.js processes)
- RDK (Note: Some environments may deploy multiple RDK servers)
- eHMP User Interface (UI)
- eHMP Balancer

The following servers are considered mock servers and are used in some environments to emulate external systems:

- VistA instances (Note: Two mock VistA instances are typically deployed: Vista Kodak and Vista Panorama)
- A central VM that holds the following mock services:
    - VE - Application Programming Interface (API) (hosts mock instances of HDR/CDS and VLER-DAS)
    - MVI
    - JMeadows

3.1.3. Ensure Properly Configured File Systems

The specific instructions for completing this step are outside the scope of the Installation Guide. Within the EDE and EO environments, file systems in the root directory cannot be added or extended. The file systems should be created and allocated to be consistent with Table 3-1.

Table 3-1 Added File Systems

VM	Added File Systems
Auditing Server	/var/chef (5GB) /var/yum_repo (5GB) /usr (variable depending upon expected data)
Solr	/var/chef (5GB) /var/yum_repo (5GB) /opt/solr (variable depending upon expected data)
JDS	/var/chef (5GB) /var/yum_repo (5GB) /usr/cachesys (variable depending upon expected data)
Persistent JDS	/var/chef (5GB) /var/yum_repo (5GB) /usr/cachesys (variable depending upon expected data)
Mock Services	/var/chef (5GB) /var/yum_repo (5GB) /usr/local (10GB)
vx-sync	/var/chef (5GB) /var/yum_repo (5GB) /opt (variable depending upon expected data) /tmp(variable depending upon expected data) /var (variable depending upon expected data)
Load Balancer	/var/chef (5GB) /var/yum_repo (5GB)
Apache Web Server	/var/chef (5GB) /var/yum_repo (5GB)
RDK	/var/chef (5GB) /var/yum_repo (5GB) /opt (variable depending upon expected data) /tmp (variable depending upon number of retained log files)

3.1.4. Ensure Default UMASK Setting

UMASK should be set to 022 in the following files under: /etc/profile.d:

- z-aitc.sh
- z-aitc.ksh
- z-aitc.csh.a.

3.1.5. Prepare User Account on Target VMs

Before deploying to a target VM, several configuration changes must be made to the user account that will be used to perform the deployment.

3.1.5.1. Setup SSH Key

Deployments are performed using rsync and Secure Shell (SSH). Therefore, an SSH key is required to authenticate access to each VM during deployment. For simplicity, it is recommended that the same SSH key is used across all VMs being deployed.

Create and set up an SSH key for the account that will be used to perform the deployment using the following steps:

1. Ensure you have an SSH key
   - You should have files on your local laptop named ~/.ssh/id_rsa and ~/.ssh/id_rsa.pub. If you do not have these, use the following commands to generate a new key pair as seen in Table 3-2:

Table 3-2 New Key Pair

Generate a New Key Pair

`mkdir ~/.ssh`
`chmod 700 ~/.ssh`
`cd ~/.ssh`
`ssh-keygen -t rsa`

2. When you're prompted to enter a file to save the key, press return and accept the default value.
   - You'll also be prompted to enter a passphrase which you'll then need to confirm. You **MUST** just press enter to create a key **WITHOUT** a passphrase.

At this point, you should have two files within the ~/.ssh folder named 'id_rsa' and 'id_rsa.pub'. These correspond to your private and public keys, respectively.

1. Copy your SSH public key to the target VMs:
   - To copy your SSH public key to the target VMs, perform the following from your local laptop as seen in Table 3-3:

Table 3-3 Copying the SSH Public Key to the Target VMs

Copying the SSH Public Key to Target VMs

```
# Ensure that you've logged onto the local VM at least once so that
# your local certificate is in the key cache
ssh s101zvg@hostname
exit

# Transfer your public SSH key to the local VM
scp ~/.ssh/id_rsa.pub s101zxx@hostname:~/id_rsa.pub

# Login to the remove server and execute the following commands
mkdir .ssh
chmod 700 .ssh
cd .ssh
touch authorized_keys
chmod 600 authorized_keys
cat ../id_rsa.pub >> authorized_keys
rm ../id_rsa.pub
```

At this point, you should be able to SSH from your deployment workspace into the VM without having to enter a password.

3.1.5.2. Configure Sudoers Access

Ensure that this user is listed within the /etc/sudoers file and that this account does NOT require a password to run sudo commands. To edit the file /etc/sudoers, visudo must be used to check for typos and correct syntax. This will ensure that the user does not get locked from accessing the server.

Typically, you want to ensure that there is a line within /etc/sudoers file for your user account that resembles the following:

- s101zxx ALL = NOPASSWD: ALL

3.2. Deployment

This section describes identifying the appropriate action for certain decision points throughout the installation process.

3.2.1. Select Deployment Package and Commit Hash

The following items must be identified to ensure that you are installing the correct version of the software:

1. Cache version number in Nexus
2. Git commit hash that corresponds to the cache version number

The cache version number should have been provided to you, and indicates the specific cache build that you are to deploy.

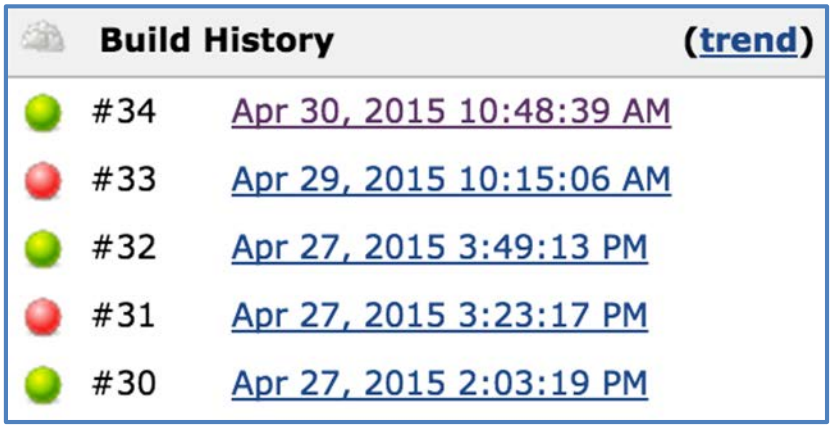
Next, determine the Git commit hash that was used to generate this cache so that we can use the same commit point within the deployment workspace. This ensures that the Vagrant file, Rake file, and other deployment artifacts are version-compatible with the cache package being deployed. Use the following steps to determine the Git commit hash for the cache build version number you were provided (it is also helpful to know the approximate date and time that the cache was created):

3. Navigate within the Jenkins server used for the CI pipeline as seen in Figure 4-1:

Figure 4-1 Jenkins Server Navigation

4. This should yield a list of the most recent cache-builder Jenkins jobs. Using the date and time the package was built, identify the Jenkins job (seen in Figure 4-2) that was used to generate the cache package:

Figure 4-2 Cache Builder Jenkins Jobs



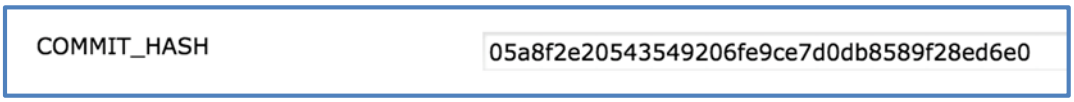
5. Click on the Jenkins cache-builder job that corresponds to the cache package to be deployed. Within the candidate job, display the console log. Toward the end of the console log, look for the entry depicted in Figure 4-3:

Figure 4-3 Jenkins Cache Builder Job Console Log

```
./chef/data_bags/sensu/ssl.json
(stdin): 3119.9 MiB / 3384.8 MiB = 0.922, 1.5 MiB/s, 37:36
uploaded to nexus -> us.vistacore.ede-eo:no-internet-cache:tar.xz:0.0.1


--------------------------------------------------------------------------------

      duration: 55 minutes and 59 seconds

--------------------------------------------------------------------------------

Performing Post build task...
Match found for :.* : True
Logical operation result is TRUE
Running script  : exit 0
[ede-eo-1.2-no-internet-cache-creator-next] $ /bin/bash -xe /tmp/hudson8500776201629739787.sh
+ exit 0
POST BUILD TASK : SUCCESS
```

6. Compare the version number shown in the Console Log to the cache version number to be deployed.
7. After verifying that the Jenkins cache-builder job matches the version of the cache package being deployed, click to view the parameters for that job. Record the value of the COMMIT_HASH parameter shown in Figure 4-4.

Figure 4-4 Jenkins Cache Builder Job Parameters



3.2.2. Prepare the Deployment Workspace

Within your deployment workspace, perform the following Git commands to ensure that you are on the Git commit point compatible with the cache package to be deployed:

1. Execute a 'Git checkout r1.2-cache-builder' to ensure that you are on the Git branch that is used for version 1.2 deployments.
2. Execute a 'git pull' to ensure that your local Git repository has all of the tags and commit hashes from origin.
3. Execute 'git checkout <commit-hash>' using the Git commit hash determined in section 4.1.

3.2.3. Generate and Modify Settings.rb File

This subsection describes how to create the deployment settings.rb file that is required during the deployment process. Each environment that gets deployed has its own version of the settings and Vagrant files. These environment-specific files are maintained within the locations listed in Table 4-1:

Table 4-4 Environment Specific File Locations

EDE	The Vagrant files are stored within the deployment workspace on the EDE Jenkins server (vaausvuxapptst36) under the directory '/var/lib/jenkins/Projects/vistacore/ehmp/infrastructure/vagrant/managed/ede-perf-test' The settings files are located within a subdirectory named '.config'.
EO	Both the Vagrant and settings files are stored within the EO Shared Resource Server (vaausehmwebprd15) under the directory '/var/ehmp-eo-shared-resources'.

The following methods are defined within the environment-specific settings files for each environment:

- **user_name:** Your user name on the VMs to be deployed.
- **private_key_path:** The full path to your SSH private key within the deployment workspace.
- **RDK**:
  - rdk1_ips: Array of IP addresses of the RDK VMs. Typically, this array should contain a single RDK IP address. When two RDK servers are used, the second server is typically specified within the 'rdk2_ips' method (see below).
  - rdk2_ips: Array of IP addresses to the RDK VMs when using a second RDK VM. When only 1 RDK VM is used, it should be specified within the 'rdk1_ips' array above and this (rdk2_ips) should be left blank.
- **machines**: Returns a list of all VMs that are deployed as part of the environment, and their dependencies. Note that the returned list excludes external servers.

Methods also exist for retrieving the IP address for each VM that is deployed. This includes the load balancer, Apache web server, vx-sync, JDS, Solr, and mock services.

These files should not change much between deployments. To create a new settings or Vagrant files, it is recommended that you start by making a copy of an existing set of files for an environment that is most similar to the new environment.

3.2.4. Set Environment Variables

Table 4-2 lists the export settings to set an environment variable:

Table 4-5 Export Settings

Export Settings	Set an environment variable named 'SETTINGS' to the full path of the environment-specific settings file (described in section 4.3) within the deployment workspace that you will be using.
Source set.env.sh	Within the deployment workspace, source the 'set.even.sh' shell script.

3.2.5. Download Deployment Package

Table 4-3 lists the directory to use when downloading the deployment package:

Table 4-6 Change Directories

Change Directories	Change to the “ehmp/infrastructure/vagrant/managed/ede-perf-test” within the deployment workspace.
Download Cache	Type: “bundle exec rake pull_cache ['version'] “and press enter”. Where version indicates the cache package version number as noted in section 4.1. NOTE: Before executing the above command for the first time, you may need to execute ‘bundle install’ from within the same directory.

3.2.6. Prepare Vagrantfile

Within the ehmp/infrastructure/vagrant/managed/ede-eo directory, there will be the default Vagrantfile that is under source control within Git. Review this Vagrantfile for any changes that have occurred since the last deployment. SourceTree’s ‘Log Selected’ function is recommended to assess the Vagrantfile for any changes that may need to be made to the environment-specific Vagrant files.

Replace the default Vagrantfile with the environment-specific Vagrantfile. Environment-specific vagrantfile for each environment are maintained within the following locations:

- EDE – On the EDE Jenkins server under ‘/var/lib/jenkins/Projects/vistacore/ehmp/infrastructure/vagrant/managed/ede-perf-test’.
- EO – On the EO Shared server (VAAUSEHMWEBPRD15) under ‘/var/ehmp-eo-shared-resources/vagrant’.

Note that in both cases, the Vagrantfile names will have a suffix that distinguishes their target environment, e.g., ‘Vagrantfile.ets’ versus ‘Vagrantfile.preprod’. However, when copied into the deployment workspace, the file must be named simply Vagrantfile.

3.2.7. VM Pre-Deployment Preparations

Certain configuration items may be overwritten when provisioning the target VMs. Therefore, configuration items should be backed up before performing the deployment or upgrade.

Perform the following steps prior to deployment:

- vx-sync
  - Backup the worker-config.json file within the /opt/vx-sync folder.
- RDK
  - Backup the /opt/rdk/config folder. Specifically, you should backup all of the files with names like ‘*config*.json’.
  - Delete the /opt/rdk folder.
  - Delete the /opt/ccow folder.
- eHMP-UI
  - Delete the /var/www/ehmp-ui/app folder.
- eHMP-BALANCER
  - Backup the /etc/httpd/sites-enabled/proxy_balancer.conf file.

3.2.8. Update Virtual Machines

The commands below must be run for each VM that will be deployed, as determined in Section 3.1.1, including both the core and mock VMs to be deployed as part of that environment. Generally speaking, the order of deployment is not important, however, it is recommended to deploy JDS, SOLR, and the mock VistA instances (when applicable) before deploying the other servers. For each VM being deployed or updated, run the following commands:

- vagrant up {machine-name}
- vagrant provision {machine-name}

Use Table 4-4 to determine the {machine-name} to use:

Table 4-7 Machine Names

Server	Machine Name
JDS	jds
Solr	solr
VistA Kodak	vista-kodak
VistA Panorama	vista-panorama
Vx-sync Server	vx-sync
Mock Services	mocks
RDK Instances	rdk1# Where '#' is 0, 1, etc for each RDK instance being deployed.
eHMP UI	ehmp-ui
eHMP Balancer	ehmp-balancer

NOTE: Although the file above is edited for the VistA connections, issues may arise depending on how the account for VistA was set up. If this is a new account, the access code needs to be changed on first entry unless the account was created with this feature turned off.

3.2.9. Validate/Restore Critical Configuration

Critical configuration files on the deployed/updated VMs must now be validated. If these are newly deployed VMs, visually verify the settings in the configuration files below. If the VMs are being updated from previous deployments, compare the updated configuration files to those that were saved in using the 'diff' command.

NOTE: Do not copy the configuration files, as there may have been changes to the baseline configuration files in the release being deployed that may be overwritten. An example of such a case would be the addition of an attribute within a JSON file.

Validate the following configuration files:

- vx-sync
    - The worker-config.json file within the /opt/vx-sync folder.

- RDK
  - Files with names like '*config*.json' within the /opt/rdk/config folder.
- eHMP-BALANCER
  - The /etc/httpd/sites-enabled/proxy_balancer.conf file.

    NOTE: If this is a new deployment, you'll need to add the following line into the proxy_balancer.conf file if it does not already exist:
  - NSSEnforceValidCerts off.

    NOTE: Typically, this is added after the existing 'NSSEngine on' setting within this file.

4. VistA KID and Site Deployment

4.1. Patch Message

The eHMP project within the VistA Evolution program will introduce expanded capabilities and modernize existing features of the VA VistA Electronic Health Record (EHR) system. The eHMP project is a multi-year effort to evolve a modern, service-oriented Enterprise Health Application Platform. This platform includes clinical knowledge enrichment and decision support services. eHMP is the Clinical data services engine, federating clinical data from a variety of VA and DoD sources into an eVPR. The eHMP framework will incorporate capabilities currently provided by JLV and HMP. eHMP will eventually replace CPRS as VA's primary point of care application. eHMP Release 1.2 Viewer Edition supports a user interface.

4.2. Patch Components

This multi-patch build consists of the patches seen in Table 5-1.

Table 5-1 Patch Components

Patch Component
MD*1.0*38
OR*3.0*390
PSB*3.0*79
TIU*1.0*106
GMRC*3.0*80
HMP 2.0

The following information is specific to the HMP 2.0 patch, which is the primary patch for eHMP. The other patches contain methods for extracting domain specific data to the eHMP and JSON Data Store.

4.3. Files and Fields Associated

File Name (#) Field Name (#) New/Modified/Deleted

--- --- ---

Table 5-2 lists the associated files and fields in the HMP 2.0 patch.

Table 5-2 Associated Files and Fields

Associated Files and Fields
HMP SUBSCRIPTION #800000
HMP PATIENT OBJECT #800000.1
HMP OBJECT #800000.11
HMP LIST #800000.2
HMP LIST DOMAIN #800000.21

Associated Files and Fields
HMP LIST ATTRIBUTE #800000.22
HMPPANEL #800001
HMP ROSTER #800001.2
HMP ACTIVITY #800001.5

4.4. Additional Information: Routines

Table 5-3 lists the informational routines.

Table 5-3 Informational Routines

Informational Routines
HMPACT>HMPAP1>HMPATRG>HMPCAC>HMPCORD>HMPCORD1>HMPCORD2>HMPCORD3 HMPCORD4>HMPCORD5>HMPCPAT>HMPCPAT1>HMPCPRS>HMPCRPC>HMPCRPC1>HMPD HMPDCRC>HMPDERRH>HMPDGMPL>HMPDGMRA>HMPDGMRC>HMPDGMV>HMPDGPF>HMPDIB HMPDJ>HMPDJ0>HMPDJ00>HMPDJ00A>HMPDJ01>HMPDJ02>HMPDJ03>HMPDJ04 HMPDJ04A>HMPDJ04E>HMPDJ05>HMPDJ05V>HMPDJ06>HMPDJ07>HMPDJ08>HMPDJ08A HMPDJ09>HMPDJ09M>HMPDJ1>HMPDJ2>HMPDJFS>HMPDJFSD>HMPDJFSG>HMPDJFSM HMPDJFSP>HMPDJFST>HMPDJT>HMPDJX>HMPDLR>HMPDLRA>HMPDLRO>HMPDMC HMPDMDC>HMPDOR>HMPDPROC>HMPDPS>HMPDPSI>HMPDPSO>HMPDPSOR>HMPDPT HMPDPXAM>HMPDPXED>HMPDPXHF>HMPDPXIM>HMPDPXSK>HMPDRA>HMPDSDAM>HMPDSR HMPDTIU>HMPDTIUX>HMPDVSIT>HMPEASU>HMPEF>HMPEFSG>HMPEFSP>HMPEFST HMPEFX>HMPEHL7>HMPELAB>HMPENSZ>HMPENSZ1>HMPEQ>HMPEQLM>HMPEQLM1 HMPEQLM2>HMPEVNT>HMPFPTC>HMPHTTP>HMPIDX>HMPJREQ>HMPJRSP>HMPJRUT HMPJSON>HMPJSOND>HMPJSONE>HMPLIST>HMPMDUTL>HMPMOCK>HMPP2I>HMPP3I HMPPANEL>HMPPARAM>HMPPATS>HMPPI>HMPPRODC>HMPPTDEMHMPPXPR1>HMPPXRM HMPROS2>HMPROS3>HMPROS4>HMPROS5>HMPROS6>HMPROS7>HMPROS8>HMPSR HMPTRPC>HMPTRPC1>HMPUPD>HMPUTILS>HMPYCSI>HMPYFRP>HMPYFRP1>HMPYFRP2 HMPYPAR

Table 5-4 details the associated options.

Forms Associated

Form Name	File #	New/Modified/Deleted
---	---	---
N/A		

Mail Groups Associated

Mail Group Name	New/Modified/Deleted
---	---
N/A	

Options Associated

Option Name Type New/Modified/Deleted

--- --- ---

Table 5-4 Associated Options

Associated Options
HMP APPLICATION PROXY
HMP APPOINTMENTS
HMP PATIENT ACTIVITY
HMP PATIENT DATA MONITOR
HMP SYNCHRONIZATION CONTEXT
HMP UI CONTEXT
HMP WB PTDEM
HMP XU EVENTS
HMPM ADD HMP PATIENT
HMPM ADD HMP USER
HMPM EMERGENCY STOP
HMPM EXTRACT MONITOR
HMPM RESTART FRESHNESS
HMPMGR

4.5. Protocols Associated

Table 5-5 lists the associated protocols.

Protocol Name New/Modified/Deleted

--- ---

Table 5-5 Associated Protocols

Associated Protocols	
HMP ADT-A04 CLIENT	HMP HL7 ADT-A04 Client
HMP ADT-A08 CLIENT	HMP HL7 ADT-A08 Client
HMP APPT EVENTS	Appointment events for HMP
HMP DG UPDATES	DG updates for HMP
HMP GMPL EVENT	Problem List events for HMP
HMP GMRA EVENTS	Allergy Events for HMP
HMP INPT EVENTS	Inpatient Movement events for HMP
HMP MDC EVENT	CLiO events for HMP

Associated Protocols	
HMP NA EVENTS	XQOR HL7 events for HMP
HMP PATIENT APPT TRIGGER	HMP Activity File Patient Appointment Trigger
HMP PCE EVENTS	PCE events for HMP
HMP PCMM TEAM	PCMM Team events for HMP
HMP PCMM TEAM POSITION	PCMM Team Position events for HMP
HMP PSB EVENTS	BCMA events for HMP
HMP XQOR EVENTS	XQOR HL7 events for HMP
HMPM EVT QUE CHANGE MAX LISTED	Change Max Limit
HMPM EVT QUE CHANGE SERVER	Change Server
HMPM EVT QUE DISPLAY DETAILS	Display Details
HMPM EVT QUE FILTER	Filter Events
HMPM EVT QUE FRESHNESS REPORT	Freshness Report
HMPM EVT QUE MGR MENU	VPR Freshness Queue Viewer
HMPM EVT QUE REFRESH	Update
HMPM EVT QUE SELECT PATIENT	Select Patient
HMPM EVT QUE SHOW TEMP GLOBALS	Temp Global Usage

4.6. Security Keys Associated

Table 5-6 lists the security keys associated.

Security Key Name

Table 5-6 Associated Security Keys

Security Key Name
HMP ADMIN
HMP EXPERIMENTAL

Templates Associated:

Template Name Type File Name (#) New/Modified/Deleted

---------- ----- ---------- ----------

N/A

New Service Requests (NSRs):

NSR #20070902

Patient Safety Issues (PSIs):

Table 5-7 lists the testing sites:

N/A

Remedy Ticket(s) & Overviews:

N/A

Test Sites:

Table 5-7 Test Sites

Test Sites
Hampton
Portland
San Diego
Loma Linda
San Antonio
Indianapolis

Documentation Retrieval Instructions:

Updated documentation describing the new functionality introduced by this patch is available.

The preferred method is to FTP[1] the files.

Directory: EHMP

Username: EHMP

Password: EHMP$STORE

Table 5-8 lists the software distribution:

Table 5-8 Software Distribution

File Name	Description	Block Size
HMP_2-0_V5-7.KID (note the version number is subject to change)	Kernel Installation & Distribution System (KIDS) Build	Blocks

[1] ftp://ftp.fo-slc.med.va.gov/

The documentation will be in the form of Adobe Acrobat files.

File Name	Description	Retrieval Format
HMP_2-0_V5-7 (note version number subject to change).KID	eHMP Project Initial Deployment	ASCII

Patch Installation:

Pre/Post Installation Overview

N/A

Installation Instructions

This patch may be installed with users on the system, although it is recommended that it be installed during non-peak hours to minimize a potential disruption. This patch should take less than five minutes to install.

Installation Procedure:

1. From the Kernel Installation and Distribution System menu, select the Installation menu. From this menu, you may elect to use the following option. When prompted for the INSTALL enter the patch number (HMPM*2.0*1):
    a. **Backup a Transport Global** - This option creates a backup message of any routines exported with the patch. It will not back up any other changes such as DDs or templates.
    b. **Compare Transport Global to Current System** - This option allows you to view all changes that will be made when the patch is installed. It compares all components of the patch (routines, DDs, templates, etc.), and backs up any other changes, such as DDs or templates.
    c. **Compare Transport Global to Current System** - This option allows you to view all changes that will be made when the patch is installed. It compares all components of the patch (routines, DDs, templates, etc.).
    d. **Verify Checksums in Transport Global** - This option allows you to ensure the integrity of the routines that are in the backup, including any other changes such as DDs or templates.
    e. **Compare Transport Global to Current System** - This option allows you to view all the changes that will be made when the patch is installed. It compares all components of this patch (routines, DDs, templates, etc.).
    f. **Verify Checksums in Transport Global** - This option allows you to ensure the integrity of the routines that are in the transport global.
2. From the installation menu, select Load a Distribution. Enter the host file location, then enter HMP_2-0_V5-6.KID

Note: The version number is subject to change.

3. From the installation menu, select the Install Package(s) option and choose the patch to install. Enter HMPM*2.0*1.
4. When prompted Want KIDS to INHIBIT LOGONs during the install? NO//, respond NO.
5. When prompted to Rebuild Menu Options YES//, respond YES.
6. When prompted Want to DISABLE Scheduled Options, Menu Options, and Protocols? NO//, respond NO.

If prompted Delay Install (Minutes): (0 - 60): 0//, respond 0. [End of file].

4.7. Site Installation Checklists

This document divides the site deployment into three distinct phases:

- Pre-Implementation/Initial Site Setup
- Pre-Production/Test Environment Implementation
- Production Environment Implementation

The checklists that detail the steps each site will need to perform in order to successfully install eHMP are located in Appendices C.1 through C.3. It should be noted that the steps may differ slightly, depending on individual site requirements.

4.8. Pre-Implementation/Initial Site Setup

Please see Appendix C.1 for the pre-implementation/initial site setup checklist with step details.

4.9. Test/Pre-Production Environment Implementation

Please see Appendix C.2 for the test/pre-production environment implementation checklist with step details.

4.10. Production Environment Implementation

Please see Appendix C.3 for the production environment implementation checklist with step details.

A. Appendix 1 - Servers for Release

Table 5-9 includes a list of the release servers. The CPU, RAM and other specifications for each server will vary from environment to environment.

Table 5-9 Release Servers

Server Type	Description
JDS	JSON Data Store
Monitoring and Performance Metrics	ElasticSearch and Kibana server
Resource Server	REST services
Solr	Apache Solr/Lucene server
vx-sync Service	Patient and operational data synchronization services
Web Server	Apache web server for HTML, CSS and JavaScript
Load Balancer	Apache load balancer for web server and Resource services

A.1. External Services

Table 5-10 is a list of the external services:

Table 5-10 External Services

Service	Description
HDR/CDS	Health Data Repository/Clinical Data Service
JMeadows	JMeadows DoD Data
MVI	Master Veterans Index
VistA	VistA instances

B. Appendix 2 – Acronyms

Table 5-11 Acronym List

Acronym	Description
ADPAC	Automated Data Processing Application Coordinator
AITC	Austin Information Technology Center
API	Application Programming Interface
ASM	ASM Research
CAC	Clinical Application Coordinator
CPRS	Computerized Patient Record System
EDE	Enterprise Development Environment
EHR	Electronic Health Record
EO	Enterprise Operations
eHMP	Enterprise Health Management Platform
FCIO	Facility Chief Information Officer
FQDN	Full Qualified Domain Name
GIT	Source Code Management and Version Control software
HMP	Health Management Platform
JDS	JSON Data Store
JSON	JavaScript Object Notation
KIDS	Kernel Installation and Distribution System
OI&T	Office of Information and Technology
SOLR	Searching On Lucene w/Replication
SSH	Secure Shell
UI	User Interface
VA	Department of Veterans Affairs
VHA	Veterans Health Administration
VistA	Veterans Health Information Systems and Technology Architecture
VM	Virtual Machine

C. Appendix 3 – Site Installation Checklists

C.1. Pre-Implementation/Initial Site Setup

The initial site setup is performed once, and will not need to be performed again for subsequent release installs.

Table 5-12 Pre-Implementation/Initial Site Setup Checklist

Task No.	Task	Responsible Party	Days Prior (-), Day of (T), and Days Post (+) Install	Pre Prod	Prod
Pre-Implementation Setup					
1	Provide a site Point of Contact (POC) list to the ASMR Release Team.	Site Office of Information & Technology (OI&T) and Veterans Health Administration (VHA) Staff	T-30	X	
2	Provide site connectivity information to the ASMR Sustainment Team.	Site/Region OI&T	T-30	X	
3	Validate station ID with site	Team ASMR	T-30	X	
4	Send the Installation Guide to the site for review.	Team ASMR	T-30	X	
5	Begin the pre-production and production account creation process for ASMR Testers.	Team ASMR	T-30	X	X
6	Confirm pre-production and production accounts for ASMR testers.	OI&T or Automated Data Processing Application Coordinator (ADPAC) (Site)	T-30	X	X
7	Validate access of the ASMR testers' accounts.	Team ASMR	T-30	X	
8	Confirm that the appropriate facility personnel have completed eHMP training.	Site Clinical Application Coordinator (CAC)	T-5	X	

Step 1: The site's POCs must be identified. The POCs should include:

- A technical person, such as the CAC
- Site or regional personnel who will be involved with the deployment
- The Facility Chief Information Officer (FCIO), who will provide the authority to deploy

Step 2: Site or regional personnel and OI&T will work together to determine site-specific information necessary for communication. The following information is needed:

- Fully Qualified Domain Name (FQDN)
- Port
- Station ID

Step 3: The station ID must be validated. This information is needed for site communication.

Step 4: Team ASMR will provide the sites with the Installation Guide for review. Site personnel will have the opportunity to ask questions about the installation process.

Step 5: Accounts for ASMR testers will be requested. Team ASMR testers will supplement the sites' test activities.

> **NOTE:** The account/access request process may differ from site to site.

Step 6: Once the all the paperwork has been filed and approved, the sites create the ASMR testers' accounts, and send access codes, via encrypted e-mail, to the designated ASMR representative.

Step 7: Testers will confirm their access.

Step 8: The site will confirm that eHMP training was completed prior to the installation of eHMP into their Test/Pre-Production environment.

C.2. Test/Pre-Production Environment Implementation

Table 5-13 is the checklist for implementation into the test/pre-production environment. Step details follow the checklist. Please note that not all steps have associates detailed information.

Table 5-13 Test/Pre-Production Environment Implementation Checklist

Task No.	Task	Responsible Party	Days Prior (-), Day of (T), and Days Post (+) Install	Pre Prod	Prod
Test/Pre-Production Environment Implementation					
1	Install Pre-Production\Test KIDS package. This will generate the test proxy\system account.	Site/Region OI&T	T	X	
2	Monitor the post-installation background jobs and global growth.	OI&T	T	X	
3	Allocate appropriate keys and menu options for end users and ASMR testers (OR/CPRS UI).	Site/Region OI&T	T	X	
4	Provide the HMP UI CONTEXT secondary menu option for end users and ASMR testers.	Site/Region OI&T	T	X	
5	Configure the eHMP server-side software (Pre-Production).	Team ASMR	T+1	X	
6	Run script on server and provide eHMP VistA ID.	OI&T			
7	Validate eHMP VistA ID.	Team ASMR	T+1	X	
8	Test connectivity to site.	Team ASMR	T+1	X	
9	Schedule and confirm time with site to perform the operational data sync.	Team ASMR		X	
10	Perform the operational data sync.	Team ASMR	T+1	X	
11	Validate the patient data synchronization.	VHA CAC/ADPAC/Informatics	T+1	X	
12	Execute the eHMP test scripts (Smoke Test).	Team ASMR	T+2	X	

Task No.	Task	Responsible Party	Days Prior (-), Day of (T), and Days Post (+) Install	Pre Prod	Prod
Test/Pre-Production Environment Implementation					
13	Execute the eHMP test scripts.	VHA CAC/ADPAC/Informatics	T+2	X	
14	Perform Pre-Production patch testing.	CAC/ADPAC/Informatics	T+2	X	
15	Perform Pre-Production functional validation.	Clinical Champion /CAC/Package ADPAC	T+3	X	

Step 1: Once the site has approved the installation of the KIDS build into their Test/Pre-Production environment, each site will be sent an e-mail with the build information and File Transfer Protocol (FTP) site location of the file. Upon the initial installation of eHMP into the Test/Pre-Production environment, a script will run, creating a proxy/system account. This step occurs only once, with the initial installation.

> **NOTE:** eHMP is a multi-build install, and for this reason we do not use Forum, as it does not support multi-build installations.

Step 2:: OI&T will provide the name of the post-installation background job to Team ASMR.

Step 3: The site must allocate the appropriate menu keys (OR/CPRS UI) for the eHMP end users and ASMR testers.

> **NOTE:** There may be additional menu options and keys for end users required by the site.

Step 4: A secondary menu key (HMP UI CONTEXT) must be assigned in order for eHMP to function properly.

Step 5: The eHMP server-side code must be configured to communicate with the site.

> **NOTE:** This activity is performed by Team ASMR.

Step 6: A script is run on the server-side to obtain the VistA ID.

> **NOTE:** This activity is performed by Team ASMR.

Step 7: The VistA ID must be validated.

> **NOTE:** This activity is performed by Team ASMR.

Step 8: Run a test to confirm connectivity to the site.

> **NOTE:** This activity is performed by Team ASMR.

Step 9: Team ASMR will schedule and confirm a time to run the operational data sync, for eHMP to compile the data from VistA.

NOTE: The initial data sync should be performed during off-peak hours, as it can take as long as four (4) hours or more, depending on the size of the site.

Step 10: Team ASMR will perform the operational data sync.

Step 11: The site CAC or ADPAC will validate the patient data sync.

Step 12: Team ASMR will run test scripts to confirm the software is performing as designed.

Step 13: The site's CAC will execute the test scripts in the Test/Pre-Production environment. The test scripts can be found on the eHMP IOC Site Coordination Materials SharePoint site[2], on the Training and Testing Materials workspace.

Step 14: The site's CAC/ADPAC will perform the Pre-Production patch testing.

Step 15: Functional validation must be conducted by site personnel after the software has been approved to go live at the site.

---

2 http://vaww.oed.portal.va.gov/pm/iehr/vista_evolution/IOC/eHMP%20IOC%20Site%20Coordination/Forms/AllItems.aspx?RootFolder=%2Fpm%2Fiehr%2Fvista%5Fevolution%2FIOC%2FeHMP%20IOC%20Site%20Coordination%2FTraining%20and%20Testing%20Materials&FolderCTID=0x012000D139DA80F8B17E46BEEB8CB9F4375531&View={DFCC967E-9113-42D6-84FA-8932F716118E}

C.3. Production Environment Implementation

Table 5-14 is the checklist for implementation into the production environment. Step details follow the checklist. Please note that not all steps have associates detailed information.

Table 5-14 Production Environment Implementation Checklist

Task No.	Task	Responsible Party	Days Prior (-), Day of (T), and Days Post (+) Install	Pre Prod	Prod
Production Environment Implementation					
1	Obtain approval to install eHMP into the Production environment from the facility's primary POC, and the facility/regional IT.	Team ASMR	T-4		X
2	Install the Production KIDS build. This will generate the test proxy\system account.	Site/Region OI&T	T		X
3	Monitor the post-installation background jobs and global growth.	OI&T	T		X
4	Allocate the appropriate keys and menu options for end users and ASMR testers (OR/CPRS UI).	Site/Region OI&T	T+1		X
5	Provide the HMP UI CONTEXT secondary menu option for end users and ASMR testers.	Site/Region OI&T	T+1		X
6	Configure eHMP software.	Team ASMR	T+1		X
7	Validate eHMP VistA ID.	Team ASMR	T+1		X
8	Test connectivity to site.	Team ASMR	T+1		X
9	Schedule and confirm time with site to perform the operational data sync.	Team ASMR	T+1		X
10	Perform the operational data sync.	Team ASMR	T+1		X
11	Validate the patient data synchronization.	CAC/ADPAC/Informatics	T+1		X
12	Execute the eHMP test scripts (Smoke Test)	Team ASMR	T+1		X

Task No.	Task	Responsible Party	Days Prior (-), Day of (T), and Days Post (+) Install	Pre Prod	Prod
Production Environment Implementation					
13	Execute the eHMP test scripts.	CAC/ADPAC/Informatics	T+2		X
14	Perform functional validation in the Production environment and go live.	Clinical Champion /CAC/Package ADPAC	T+3		X

Step 1: Once the site has completed testing, Team ASMR will obtain approval from the facility's primary POC, and the facility/region IT. Once approved, Team ASMR will schedule the installation of the software into the Production environment.

Step 2: Once the site has approved the installation of the KIDS build into their Test/Pre-Production environment, each site will be sent an e-mail with the build information and File Transfer Protocol (FTP) site location of the file. Upon the initial installation of eHMP into the Test/Pre-Production environment, a script will run, creating a proxy/system account. This step occurs only once, with the initial installation.

> **NOTE:** eHMP is a multi-build install, and for this reason we do not use Forum, as it does not support multi-build installations.

Step 3: OI&T will provide the name of the post-installation background job to Team ASMR.

Step 4: The site must allocate the appropriate menu options and keys (OR/CPRS UI) for the eHMP end users and ASMR testers.

> **NOTE:** There may be additional menu options and keys for end users required by the site.

Step 5: A secondary menu option HMP UI CONTEXT must be assigned for eHMP to function properly.

Step 6: The eHMP server-side code must be configured to communicate with the site.

> **NOTE:** This activity is performed by team ASMR.

Step 7: A script is run on the server-side to obtain the VistA ID, and the VistA ID is validated.

> **NOTE:** This activity is performed by team ASMR.

Step 8: A test will be performed to confirm connectivity to the site.

> **NOTE:** This activity is performed by team ASMR.

Step 9: Team ASMR will schedule and confirm a time to run the operational data sync, for eHMP to compile the data from VistA.

> **NOTE:** The initial data sync should be performed during off-peak hours, as it can take as long as four (4) hours or more, depending on the size of the site.

Step 11: The site CAC or ADPAC will validate the patient data sync.

Step 12: Team ASMR will run test scripts to confirm that the software is working as designed.

Step 13: The site's CAC will execute the test scripts in the Production environment. The test scripts can be found on the eHMP IOC Site Coordination Materials SharePoint site[3], on the Training and Testing Materials workspace.

Step 14: Functional validation must be conducted by site personnel after the software has been approved to go live at the site.

---

3 http://vaww.oed.portal.va.gov/pm/iehr/vista_evolution/IOC/eHMP%20IOC%20Site%20Coordination/Forms/AllItems.aspx?RootFolder=%2Fpm%2Fiehr%2Fvista%5Fevolution%2FIOC%2FeHMP%20IOC%20Site%20Coordination%2FTraining%20and%20Testing%20Materials&FolderCTID=0x012000D139DA80F8B17E46BEEB8CB9F4375531&View={DFCC967E-9113-42D6-84FA-8932F716118E}