

# VAR-Resources 4.2.x - Installation Instructions

Icon

## Assumptions

- Operations personnel is familiar with JEE applications and knows how to deploy applications.
- Operations personnel knows how to take backups and revert deployments if there are failures.
- Operations personnel knows how to procure environment-specific property values.
- IP addresses are interchangeable with fully qualified domain names.

Pre-requisites:

1. APIGateway, Consul, and other PPG components.
2. Containers above must already be running in the swarm.

Associated Project: veteran-appointment-requests

Priority: Standard

Components: DTR, Nexus, Consul, MongoDB, Oracle

Installation:

Jenkins Job already created at: <http://dockerdev.vamiscece.io:8080/job/var-resources-app-build/>

DTR registry needs to exist at: [dtr.vamiscece.io/vamis/var-resources](http://dtr.vamiscece.io/vamis/var-resources)

Nexus registry'ssccs' needs to exist in: [dockerdev.vamiscece.io:9250/nexus](http://dockerdev.vamiscece.io:9250/nexus)

### Service/Application to be Deployed

Service	Branch	URL
VarMessagingMicroservice	Release 2.0.1	<a href="https://coderepo.mobilehealth.va.gov/projects/VAR/repos/var-messaging-microservice/browse?at=refs%2Fheads%2FRelease-2.0.1-Branch">https://coderepo.mobilehealth.va.gov/projects/VAR/repos/var-messaging-microservice/browse?at=refs%2Fheads%2FRelease-2.0.1-Branch</a>
Facility Resources	Release 2.0.1	<a href="https://coderepo.mobilehealth.va.gov/projects/VAR/repos/facility-resources/browse?at=refs%2Fheads%2FRelease-2.0.1-Branch">https://coderepo.mobilehealth.va.gov/projects/VAR/repos/facility-resources/browse?at=refs%2Fheads%2FRelease-2.0.1-Branch</a>

Consul keys:

```
export VAMF_ENVIRONMENT=ece-dev
export CONSUL_HTTP_ADDR=vhaecedevucp04.vamiscece.io:8500
```

Add the following section of code to the docker-compose.yml file:

...

```
var-resources:
image: var-resources
links:
- var-messaging-microservice
depends_on:
- var-messaging-microservice
ports:
- "9092:8080"
environment:
- JAVA_OPTS=-XX:+CMSClassUnloadingEnabled -Xrs -Dcom.sun.xml.ws.transport.http.client.HttpTransportPipe.dump=true
...
```

Run the following commands to register the service with Consul:

```

...
curl -H "X-Consul-Token: $CONSUL_TOKEN" -s -X PUT $CONSUL_HTTP_ADDR:8500/v1/kv/vamf/$VAMF_ENVIRONMENT/apigateway/1.0
/services/var-resources -d '{"location":"/var/VeteranAppointmentRequestService","service":"var-resources-8080","redirect":"off","headers":{"X-
Real-IP":"$remote_addr"}}' > /dev/null
...

```

This document outlines the instructions for installing the **VeteranAppointmentRequestService v4.2.x**.

<b>Service Name</b>	Veteran Appointment Request Service
<b>Enclave</b>	Veteran Only
<b>Docker Swarm</b>	VarResources
<b>Java Runtime</b>	JRE 1.8

## Service and Data Source Dependencies

**VeteranAppointmentRequestService v4.2.x** depends on a number of data sources and services described below:

Dependency	Type	Dependency Type	Description
MDWS v3.2.8	Service	Runtime	Verify this service is deployed and available/accessible for Veteran Appointment Request Service. The endpoint configuration is described in the Property override section.
MVI	Service	Runtime	Verify this service is deployed and available/accessible for Veteran Appointment Request Service. The endpoint configuration is described in the Property override section ( <b>mvi.properties</b> )
FacilityService	Service	Runtime	Verify this service is deployed and available/accessible for Veteran Appointment Request Service. The endpoint configuration is described in the Property override section ( <b>facility.properties</b> )
VarMessagingMicroservice	Service	Runtime	Verify this service is deployed and available/accessible for Veteran Appointment Request Service. The endpoint configuration is described in the Property override section ( <b>vm-m-config.properties</b> )
MongoDB	Data Source (MongoDB Database)	Runtime	Verify this data source exists and is accessible. Otherwise create it.
VARDB	Data Source (Oracle Database)	Deployment and Runtime	Verify this data source and JNDI exists. Otherwise, create it.

## Property Overrides

**VeteranAppointmentRequestService v4.2.x** all of these properties are now stored as environment variables, please make the desired changes as follows for each component of VAR Resources.

**VeteranAppointmentRequestService v4.2.x** uses **envconsul** to pull the environment variables down from consul at the path **appconfig/{VAMF\_ENVIRONMENT}/var-resources**. The available consul keys are listed below.

## SSL CertConfigutaion

In order to support 2-way SSL with MVI, Mongo, and Oracle - an SSL cert needs to be imported into the keystore. The following properties need to be added to consul in order to have the cert automatically imported at startup.

## ssl cert configuration

```
SVC_KEY_PKCS12=<base64 encoded key cert -- as a chain cert>
SVC_KEY_PKCS12_PWD=<the password of the cert>
SVC_KEYSTORE_PWD=<the keystore password>
```

## VARDB Data Source Configuration

### vardb environment variables

```
vardb_dialect=org.hibernate.dialect.Oracle10gDialect
vardb_driverClassName=oracle.jdbc.OracleDriver
vardb_url=jdbc:oracle:thin:@<oracle_ip/hostname>:<oracle_port>:<sid>
vardb_user=<username>
vardb_password=<password>
```

## MDWS Configuration

### mdws environment variables

```
scheduling.endBasePointUrl=<MDWS end base point URL>
scheduling.endpointUrl=SchedulingSvc.asmx
```

## CDW Data Source Configuration

### cdw environment variables

```
cdw_dialect=org.hibernate.dialect.SQLServer2008Dialect
cdw_driverClassName=net.sourceforge.jtds.jdbc.Driver
cdw_url=jdbc:jtds:sqlserver://<cdw_ip/hostname>:<cdw_port>
cdw_user=<username>
cdw_password=<password>
```

## MVConfiguration

MVluses2-way SSL. Please make sure to exchange the certificates between MVlandServer hosted with the **VeteranAppointmentRequestService v4.2.x**. The SSL configuration is only needed in Pre-Prod and Production as of Jun 2017.

### mvi environment variables

```
mock.mvi=false # should stay false; true is reserved for development
mvi.endpointUrl=<mvi web service URL>
```

## MongoDB Configuration

### mongo environment variables

---

```
#IP addresses are interchangeable with fully qualified domain names.
mongo.host=<Comma separated MongoNodeIP/hostname:port> //Example: For replica sets, mongo.host should
be set as mongol:port1,mongo2:port2,mongo3:port3
mongo.user=<var service-specific account> (See PMDS ticket for this service)
mongo.secretkey=<password>
mongo.database.name=var-utility
mongo.ssl=true
mongo.useAuthentication=true
```

## Facility Timezone Configuration

### facility environment variables

```
facility.timezone.fetch.all.uri=<environment-address>/FacilityService/v2/rest/public/facility/timezone
facility.timezone.fetch.by.id.uri=<environment-address>/FacilityService/v2/rest/public/facility/
{facility-id}/timezone
```

## Var Api Keys Configuration

### var api keys environment variables

```
apiKeys=googleApiKey:
<myGoogleApiKey>
```

More apiKeys can be added to this configuration through the same line.

```
apiKeys=googleApiKey:<myGoogleApi
Key>,testkey:testValue
```

## Recall Reminders Configuration

### recall-reminders environment variables

```
# run at 6:00am every day
batch.recall.reminders.cron.expression=0 0 6 * * *

vmm.token.uri=<var-messaging-microservice host>/VarMessaging/v2/rest/api/auth
vmm.post.recall.reminder.uri=<var-messaging-microservice host>/VarMessaging/v2/rest/api/message
recall.num.days = 45
var.secret=VARSECRET

recall.reminder.email.subject=It is time to schedule a Follow-Up Appointment at <Clinic Name>

recall.reminder.email.body=\
<p>Your provider has requested that you schedule a follow up appointment around <recall date>. Please
contact the facility below to make an appointment:</p>\
<p>\
<Facility Name>\
<br /><Clinic Name>\
<br /><Clinic location friendly name>\
<br /><Provider Name>\
</p>\
<p><a href="https://www.va.gov/directory/guide/division.asp?dnum=1">View VA Facilities</a></p>\
<p>Your VA Healthcare Team</p>\

vmm.inapp.notification.fetch.uri=<var-messaging-microservice host>/VarMessaging/v2/rest/api/inApp/
{assigningAuthority}/{uniqueId}
```

```
vmm.inapp.notification.mark.read=<var-messaging-microservice host>/VarMessaging/v2/rest/api/inApp/{messageIdentifier}/read
vmm.inapp.notification.delete.uri=<var-messaging-microservice host>/VarMessaging/v2/rest/api/inApp/{messageIdentifier}
```

## Enable Primary Care direct scheduling

Use mongo shell to enable direct scheduling of Primary Care appointments. In VAR version 4.2.4 and VAR Utility version 2.1.1, Primary Care direct scheduling is no longer enabled by default, but requires explicit configuration for sites which support it. The script below will ensure that Primary Care direct scheduling remains enabled for those facilities which do not already have this setting applied. Refer to MongoDB documentation for syntax of the command to execute a script against the the var-utility database specified. A sample partial command is provided below. Please issue the complete command for the environment being installed.

```
mongo --ssl -u <username> -p <password> --authenticationDatabase <authentication database> var-utility set-
primary-care-support.js
```

Download the the script: [set-primary-care-support.js](#)

## Import Community Care Types of Care (if it has not been done)

Use **mongoimport** utility to import the list of Types of Care into the MongoDB instance configured above. Import data into the **communityCareTypeOfCares** collection in the **var-utility** database. Refer to MongoDB documentation for syntax of the command to import a json file into the designated collection in the designated database. A sample partial command is provided below. Please issue the complete command for the environment being installed.

```
mongoimport --ssl --db var-utility --collection communityCareTypeOfCares --drop --jsonArray --file <community-care-type-of-cares.json> -u <username> -p <password> --authenticationDatabase <authentication database>
```

### community-care-types-of-care.json

```
[
  {
    "_id" : "CCAUDHEAR",
    "name" : "Audiology (hearing aid support)",
    "details" : "This is an office visit for Veterans who already have a hearing aid and who need assistance with the device. This visit is for trouble shooting or adjusting your hearing aid for best performance. This may include technical adjustments, cleaning, and/or reprogramming for new listening situations and environments. This is not a visit to initially be evaluated for or to obtain a hearing aid.",
    "confirmation" : "This request is for hearing aid support only. If you have other needs, please contact your Primary Care Provider to discuss the issue. If you are distance eligible and do not have a Primary Care Provider, you can use this app to request help in getting one assigned."
  },
  {
    "_id" : "CCAUDRTNE",
    "name" : "Audiology (routine hearing exam)",
    "details" : "This includes an office visit for a hearing exam and evaluation using non-invasive tests to check your hearing and inner ear health. The exam may lead to prevention and treatment recommendations, a request for further consultation, and/or a recommendation for hearing aids. A routine exam is not for any new or sudden changes with your hearing or ears.",
  }
]
```

```

        "confirmation" : "This request is for routine hearing exam
only. If you have other needs, please contact your Primary Care Provider
to discuss the issue. If you are distance eligible and do not have a
Primary Care Provider, you can use this app to request help in getting one
assigned."
    },
    {
        "_id" : "CCOPT",
        "name" : "Optometry (routine eye exam)",
        "details" : "This includes an office visit for an eye exam
and vision evaluation, which may include dilating your eyes. The exam may
lead to prevention and treatment recommendations, a request for further
consultation, and/or a prescription for eye glasses or contact lenses. A
routine exam is not for any new or sudden changes with your vision or
eyes.",
        "confirmation" : "This request currently only supports
routine eye exams. If you have other needs, please contact your Primary
Care Provider to discuss the issue. If you are distance eligible and do
not have a Primary Care Provider, you can use this app to request help in
getting one assigned."
    },
    {
        "_id": "CCPCPRVDR",
        "name": "Primary Care (find a provider)",
        "details": "TBD",
        "confirmation": "TBD"
    },
    {
        "_id": "CCPOD",
        "name": "Podiatry",
        "details": "TBD",
        "confirmation": "TBD"
    }
}
]

```

## Create Community Care Appointment Request (if it has not been done)

Run the following SQL script on **VARDB** schema.

### VAR-9277.sql

```

CREATE TABLE "VARDB"."CC_APPOINTMENT_REQUEST"
(
    "CC_APPOINTMENT_REQUEST_ID" VARCHAR2(32 BYTE) NOT NULL ENABLE,
    "APPOINTMENT_REQUEST_ID" VARCHAR2(32 BYTE) NOT NULL ENABLE,
    "TRAVEL_BURDEN" NUMBER NOT NULL ENABLE,

```

```

"UNUSUAL_TRAVEL_BURDEN" NUMBER,
"DISTANCE_WILLING_TO_TRAVEL" NUMBER NOT NULL ENABLE,
"ADDRESS" VARCHAR2(100 BYTE),
"CITY" VARCHAR2(50 BYTE),
"STATE" VARCHAR2(2 BYTE),
"ZIPCODE" VARCHAR2(5 BYTE),
"PREFERRED_ZIPCODE" VARCHAR2(5 BYTE) NOT NULL ENABLE
);

```

## Update Community Care Appointment Request in Oracle DB (if it has not been done)

Run the following SQL script on VARDB schema.

### vardb\_added\_col\_known\_eligibility.sql

```

ALTER TABLE CC_APPOINTMENT_REQUEST
MODIFY (
    TRAVEL_BURDEN NULL
);
ALTER TABLE CC_APPOINTMENT_REQUEST
ADD (
    DISTANCE_ELIGIBLE NUMBER
);
UPDATE CC_APPOINTMENT_REQUEST
SET DISTANCE_ELIGIBLE = 0;
ALTER TABLE CC_APPOINTMENT_REQUEST
MODIFY (
    DISTANCE_ELIGIBLE NOT NULL
);

```

## Update Appointment Request in Oracle DB (if it has not been done)

Run the following SQL script on VARDB schema.

### SCV-2857.sql

```

ALTER TABLE VARDB.APPOINTMENT_REQUEST
ADD (
    LOCATION_ID VARCHAR2(100),
    AUTHORITATIVE_NAME VARCHAR2(255),

```

```
FRIENDLY_NAME VARCHAR2(255)
);
COMMIT;
```

## Update Appointment Request in Oracle DB (if it has not been done)

Run the following SQL script on VARDB schema.

```
ALTER TABLE VARDB.APPOINTMENT_REQUEST DROP COLUMN LOCATION_ID;
ALTER TABLE VARDB.APPOINTMENT_REQUEST DROP COLUMN AUTHORITATIVE_NAME;

-- Make sure the primary key is set to the facility code. Previously it
was the name.
ALTER TABLE VARDB.FACILITY DROP CONSTRAINT PK_FACILITY;
DROP INDEX VARDB.PK_FACILITY;
ALTER TABLE VARDB.FACILITY ADD CONSTRAINT PK_FACILITY PRIMARY KEY
(FACILITY_CODE);
```

## Disabling Features

**VeteranAppointmentRequestService** has the ability to disable features based on feature set. This release includes Community Care features as well as Telehealth features, disabled by default. This is controlled by a properties file:

var-disabled-features.properties and the property `disabledFeatures.disabledFeatures`. The default setting is below:

### var-disabled-features.properties

```
#disabled features for var
disabledFeatures.disabledFeatures=CC_FEATURE_SET,TH_FEATURE_SET
```

To enable specific features, the feature set flag entry must be removed from the comma-separated list (the flags can be in any order):

```
#example: disables both Community Care features and Telehealth features
for VAR
disabledFeatures.disabledFeatures='TH_FEATURE_SET,CC_FEATURE_SET'

#example: enables just Community Care features for VAR
disabledFeatures.disabledFeatures='TH_FEATURE_SET'

#example: enables just Telehealth features for VAR
disabledFeatures.disabledFeatures='CC_FEATURE_SET'

#example: enables all optionally configurable features for VAR
disabledFeatures.disabledFeatures=''
```



# CDW Check List

Ensure that the **CDW** instance connected to this installation has the correct version of the following stored procedures and functions:

1. `OIA_MobileHealth.App.ap_GetPtActiveTeamsMembers_v4`
2. `OIA_MobileHealth.App.ap_GetPtActiveTeamsClinics_v5`
3. `OIA_MobileHealth.App.ap_GetActiveClinics_v3`
4. `OIA_MobileHealth.App.fn_VAR_BuildStopcodeTable`
5. `OIA_MobileHealth.App.ap_ApptRecallRemindersByDate_v1`
6. `OIA_MobileHealth.App.ap_GetChildFacilities_v1`

## Important Note About Recall Reminders

**VAR Resources 4.0+** is configured with the Recall Remindercron job turned on by default. This means that every Docker container deployed will run the job and duplicate Recall Reminders will go out. To remediate this issue, it is advised that VAR Resources only be deployed on a single Docker container, until the worker service can be deployed to its own container. In the event that more VAR Resources Docker containers are needed for load balancing, be sure to turn off the cron job by setting the time to run to the past:

**recall-reminders.properties (or via Consul)**

```
batch.recall.reminders.cron.expression=0 0 6 * * * 2016
```

## AppDynamics Configuration

Taken from [DRAFT - How to integrate AppDynamics agent with Tomcat](#)

### 1. Modify your app's docker-compose.yml with Environment Variables for Consul

Add `CONSUL_*` Variables to Environment

This image requires the environment variable `CONSUL_APPDYNAMICS_PATH`, `CONSUL_SERVER`, `CONSUL_TOKEN`, and `CONSUL_PATH` to be set. Please add the following as an environment variables in the service's docker-compose.yml:

environment:

- `"CONSUL_SERVER=${CONSUL_HTTP_ADDR}"`
- `"CONSUL_TOKEN=${CONSUL_MASTER_TOKEN}"`
- `"CONSUL_PATH=appconfig/${VAMF_ENVIRONMENT}/var-resources"`
- `"CONSUL_APPDYNAMICS_PATH=appconfig/${VAMF_ENVIRONMENT}/appdynamics"`

### 2. Set **CONSUL\_PATH** key/values

Consul is used to set properties for the AppDynamics Agent. Please set the following values in the applications Consul path. `appconfig/${VAMF_ENVIRONMENT}/var-resources/*`

`APPDYNAMICS_MONITORING_ENABLED`

`APPDYNAMICS_APPLICATION_NAME`

`APPDYNAMICS_NODE_PREFIX`

### 3. Enable AppDynamics Agent

Consul is used to enable/disable AppDynamics Agent.

### Set APPDYNAMICS\_MONITORING\_ENABLED in application's Consul path

set APPDYNAMICS\_MONITORING\_ENABLED to true in the application's Consul config to enable AppDynamics Agent.

```
appconfig/${VAMF_ENVIRONMENT}/<YOUR_APP>/APPDYNAMICS_MONITORING_ENABLED=true
```

### Set APPDYNAMICS\_APPLICATION\_NAME in application's Consul path (*required if APPDYNAMICS\_MONITORING\_ENABLED=true*)

set APPDYNAMICS\_APPLICATION\_NAME to the application name in the application's Consul config.

```
appconfig/${VAMF_ENVIRONMENT}/var-resources/APPDYNAMICS_APPLICATION_NAME="VAMF <VAMF_ENVIRONMENT> var-resources"
```

### Set APPDYNAMICS\_NODE\_PREFIX in application's Consul path(*required if APPDYNAMICS\_MONITORING\_ENABLED=true*)

set APPDYNAMICS\_NODE\_PREFIX to the application node prefix in the application's Consul config.

```
appconfig/${VAMF_ENVIRONMENT}/<YOUR_APP>/APPDYNAMICS_NODE_PREFIX=var-resources
```

#### 4. Set CONSUL\_APPDYNAMICS\_PATH key/values (GLOBAL) (if APPDYNAMICS\_MONITORING\_ENABLED=true):

Consul is used to set properties for the AppDynamics Agent. Please ensure the following values are set in the "appdynamics" Consul path. These are set once for all apps.

```
appconfig/${VAMF_ENVIRONMENT}/appdynamics/*
```

CONTROLLER\_HOST

ACCOUNT\_NAME

ACCOUNT\_ACCESS\_KEY

## Deployment

**VeteranAppointmentRequestService v4.2.x** is a standard VAMF web service that has been developed and tested on the ECE environment. After verifying the dependencies and property configuration, build the docker image from the **JENKINS Location** from the VDD page on the Docker Container.

Consul keys:

```
export VAMF_ENVIRONMENT=ece-dev
export MONGO_USER=<ops provided username>
export MONGO_PASS=<ops provided password>
export MONGO_HOST=mongo-di-n5.mobilehealth.va.gov
export CONSUL_HTTP_ADDR=vhaecdevucp04.vamiscece.io:8500
```

```
curl -H "X-Consul-Token: $CONSUL_TOKEN" -s -X PUT $CONSUL_HTTP_ADDR/v1/kv/vamf/${VAMF_ENVIRONMENT}
/apigateway/1.0/services/var-resources -d '{"location":"/var/VeteranAppointmentRequestService", "
service":"var-resources", "redirect":"off", "headers":{"X-Real-IP":"$remote_addr"}}'
```

## Validation

Ensure that <https://<environment-address>/var/VeteranAppointmentRequestService> returns "Welcome to VeteranAppointmentRequestService" with the correct manifest.

## Migrate existing Patient/Appointment data to ICN format without hash (new for 4.2.7) *(Maintenance Window installation only)*

**\*\*\*This step will change existing data to standardize primary key data formats. This is a required step in order for 4.2.7 to work. Without this step, some appointments may not display for MHV-only users or for DSLogon users who also use MHV.\*\*\***

**Please ensure there is a backup or snapshot of the existing database before running the script, and that this migration is done during a maintenance window**

Run the following script against Oracle VARDB: <https://coderepo.mobilehealth.va.gov/projects/VAR/repos/var-resources/browse/var-db/sql-release-scripts/removeIcnHash.sql?at=refs%2Fheads%2Frelease%2F4.2>

## Rollback Procedure (v4.2.x to v4.1.4)

### Mongo Rollback

Drop the **communityCareTypeOfCares** collection in the **var-utility** database. A command is provided below after being authenticated with mongo.

```
db.communityCareTypeOfCares.drop();
```

### Oracle Rollback

**\*\*\*New for 4.2.7\*\*\* *(Maintenance Window)***

To perform a complete rollback as of changes made to 4.2.7 Install Instruction set, restore the database from the snapshot taken prior to the installation

### Prior to 4.2.7

Drop the CC\_APPOINTMENT\_REQUEST table in **VARDB** schema.

#### Drop CC\_APPOINTMENT\_REQUEST

```
DROP TABLE CC_APPOINTMENT_REQUEST;
```

For rolling back changes to the VAR\_APPOINTMENT\_REQUEST table in **VARDB** schema, you can either logically delete the columns or physically delete.

#### Logical Delete

```
ALTER TABLE table_name SET UNUSED (LOCATION_ID, AUTHORITATIVE_NAME,
FRIENDLY_NAME);
COMMIT;
```

### Physical Delete

```
ALTER TABLE table_name DROP (LOCATION_ID, AUTHORITATIVE_NAME,  
FRIENDLY_NAME);  
COMMIT;
```

Remove the Docker container and follow the Install Instructions for version 4.1.4, which is the version currently installed and operational in production. See [VAR 4.1 install instructions](#). Note: For national release the versions VAR Web 4.1.4 + 37 and VAR Resources 4.1.4+59.