

# Scheduling Manager Resources 2.2.x - Installation Instructions

- Assumptions
- Summary
- Service and Data Source Dependencies
- Property Overrides
  - Disabling Features
  - VARDB Data Source Configuration
  - AUTHDB Data Source Configuration
  - MDWS Configuration
  - Authorization Services and OAuth Configuration
  - MVI Configuration
  - Metrics Queue Configuration
  - Email Notification Configuration
  - Scheduling Manager OAuth Configuration
  - MongoDB Configuration
- Import Community Care Types of Care (if it has not been done)
- Create Community Care Appointment Request (if it has not been done)
- Update Community Care Appointment Request in Oracle DB (if it has not been done)
- Update Appointment Request in Oracle DB (if it has not been done)
- Insert new Detail Codes in Oracle DB (if it has not been done)
- Deployment
- Proxy Configuration
- Validation
- Rollback procedures (v2.2.3 to v2.0.3)

## Assumptions

- Operations personnel is familiar with JEE applications and knows how to deploy applications into a WebLogic environment.
- Operations personnel knows how to take backups and revert deployments if there are failures.
- Operations personnel knows how to procure environment-specific property values.

## Summary

This document outlines the instructions for installing the **SchedulingManagerService**.

<b>Service Name</b>	Scheduling Manager Service
<b>Enclave</b>	Staff Only
<b>WebLogic Target</b>	SchedulingManagerServices
<b>Java Runtime</b>	JRE 1.7

## Service and Data Source Dependencies

**SchedulingManagerService** depends on a number of data sources and services described below:

Dependency	Type	Dependency Type	Description
VARDB	Data Source (Oracle Database)	Deployment and Runtime	Verify this data source and JNDI exists. Otherwise, create it.
AUTHDB	Data Source (Oracle Database)	Runtime	Verify this data source and JNDI exists. Otherwise, create it.
Authorization Services v4.3.x	Service	Runtime	Verify this service is deployed on the cluster.

MDWS v3.2.8	Service	Runtime	Verify this service is deployed and available/accessible for Scheduling Manager Resources. See configuration section for property configuration
MVI	Service	Runtime	Verify this service is deployed and available/accessible for Scheduling Manager Resources. See configuration section for property configuration
Staff Video Connect Service v1.0.0	Service	Runtime	Provide ability to schedule patient Video Visit appointments.
Video Visit Service v1.0.2	Service	Runtime	Provide patient Video Visit appointments created by vmrlink restful service.
MongoDB	Data Source (MongoDB Database)	Runtime	Verify this data source exists and is accessible. Otherwise create it.
SSOE Proxy	Service	Runtime	If using a mock login provider for testing, verify version 1.0.0 is installed in the WebLogic staff cluster
User Context	Service	Runtime	Verify this service is deployed and available/accessible for Scheduling Manager Resources. See configuration section for property configuration
Health Adapter 3.4.1	Service	Runtime	Verify this service is deployed and available/accessible for Scheduling Manager Resources. See configuration section for property configuration

## Property Overrides

**SchedulingManagerService** uses Spring Cocoon configurator and Spring property manager to configure property files specified in the classpath.

1. Verify that /opt/applications/**haconfig** folder exists and is specified in the class path using Web Logic Node Server startup configuration. Otherwise, create the folder and add it to the classpath.
2. Verify that /opt/applications/**haconfig/SchedulingManagerService/v2m2px/properties (update x to current version i.e v2m2p2)** folder exists, otherwise create it.

Configure dependencies as described below: Each section should be added in the specific property file to facilitate property management.

These files should be created under /opt/applications/**haconfig/SchedulingManagerService/v2m2px/properties (update x to current version i.e v2m2p2)**

## Disabling Features

**SchedulingManagerService** has the ability to disable features based on feature set. This release includes Community Care features that are disabled by default. This is controlled by a properties file:

scheduling-manager-disabled-features.properties and the property **disabledFeatures.disabledFeatures**. The default setting is below:

### scheduling-manager-disabled-features.properties

```
#disabled features for scheduling manager
disabledFeatures.disabledFeatures=CC_FEATURE_SET
```

To enable Community Care features, remove the CC\_FEATURE\_SET property in the disabledFeatures in haconfig and restart Scheduling Manager Resources in Weblogic.

#### **scheduling-manager-disabled-features.properties**

```
#enables Community Care features for Scheduling Manager
disabledFeatures.disabledFeatures=
```

## **VARDB Data Source Configuration**

#### **vardb.properties**

```
vardb_jndiname=VARDB
dataSource.ref=jndiDataSource
vardb_dialect=org.hibernate.dialect.Oracle10gDialect
```

## **AUTHDB Data Source Configuration**

#### **authdb.properties**

```
authdb_jndiname=AUTHDB
authdb_dialect=org.hibernate.dialect.Oracle10gDialect
```

## **MDWS Configuration**

#### **mdws.properties**

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

## **Authorization Services and OAuth Configuration**

#### **oauth.properties**

```
oauth.validateTokenUri=https://<environment-address>/AuthorizationServices
/rest/validateToken
```

## **MVI Configuration**

MVI uses 2-way SSL. Please make sure to exchange the certificates between MVI and Server hosted with the **SchedulingManagerService**. The SSL configuration is only needed in Pre-Prod and Production as of Jun 2017.

#### **mvi.properties**

```
mvi.endpointUrl=<mvi web service URL>
mvi.readTimeout=600000
mvi.connectionTimeout=50000
mvi.senderId=200CIH
mvi.keystorePath=<key store path>
mvi.keystorePassword=<key store password>
mvi.keyPassword=<private key password>
mvi.truststorePath=<trust store path>
mvi.truststorePassword=<trust store password>
```

## Metrics Queue Configuration

### metrics.properties

```
# Beanstalk configuration used by Metrics Services
beanstalk.host=<beanstalk ip address>
beanstalk.port=<beanstalk port>

# Metrics queue configuration used by Metrics Services
metrics.beanstalk.enabled=true
metrics.queue.name=metricsServiceQueue
```

## Email Notification Configuration

### email-notification.properties

```
email.notifications.fromAddress=no-reply@vha.gov
email.notifications.replyToAddress=no-reply@vha.gov
```

## Scheduling Manager OAuth Configuration

The **client Id** and **Secret** should be registered with **Authorization Services**. Registration should follow **Authorization Services** specific instructions.

### scheduling-manager-prod-oauth.properties

```
oauthDetails.clientId=schedulingManager
oauthDetails.clientSecret=SCHEDULINGMANAGER
oauthDetails.redirectUri=<environment address>/scheduling-manager
oauthDetails.tokenUrl=<environment address>/AuthorizationServices/oauth
/token
oauthDetails.scope=read
```

## MongoDB Configuration

### mongo.properties

```
#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=<scheduling-service-specific account> (see PMDS ticket for this
service)
mongo.secretkey=<password>
mongo.database.name=var-utility
mongo.ssl=true
mongo.useAuthentication=true

facility.mongo.host=<Comma separated MongoNodeIP/hostname:port>
//Example: For replica sets, mongo.host should be set as mongol:port1,
mongo2:port2,mongo3:port3
facility.mongo.user=<facility-service-specific account> (see PMDS ticket
for this service)
facility.mongo.secretkey=<password>
facility.mongo.database.name=facility
facility.mongo.ssl=true
facility.mongo.useAuthentication=true
```

## 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 <communi
ty-care-type-of-cares.json> -u <username> -p <password> --authenticationDatabase <authentication database>
```

### community-care-types-of-cares.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 app 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": "CCPOD",
        "name": "Podiatry (routine exam)",
        "details": "This includes an office visit for an exam of
the foot, ankle and related structures of the leg. The exam may lead to
diagnosis and treatment to include common conditions like callous,
bunions, and other conditions requiring in office treatment, orthotics,
and possibly the need for surgery. If you are experiencing any acute
conditions or sudden onset of pain, break in the skin with possible
infection, please contact your local VAMC.",
        "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.

If you are using a version of Scheduling Manager greater than 2.2.1 this has to be done

### 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;

```

## Insert new Detail Codes in Oracle DB (if it has not been done)

If you are using a version of Scheduling Manager greater than 2.2.1 this has to be done

### SCV-2955.sql

```

INSERT INTO "VARDB"."DETAIL_CODE" (DETAIL_CODE_ID, PROVIDER_MESSAGE,
VETERAN_MESSAGE) VALUES ('DETCODE15', 'Scheduled in Community on requested
date.', 'Your Community appointment was scheduled based upon your
request.');
```

```

INSERT INTO "VARDB"."DETAIL_CODE" (DETAIL_CODE_ID, PROVIDER_MESSAGE,
VETERAN_MESSAGE) VALUES ('DETCODE16', 'Scheduled in Community on alternate
date.', 'Your Community appointment has been scheduled at a time that
differs from what you originally requested. Please log in to VA
Appointments to review your appointment dates and times.');
```

### SCV-2954.sql

```

INSERT INTO "VARDB"."DETAIL_CODE" (DETAIL_CODE_ID, PROVIDER_MESSAGE,
VETERAN_MESSAGE) VALUES ('DETCODE17', 'Cancelled at Veteran request',
'Your appointment request has been cancelled at your request.');
```

```

INSERT INTO "VARDB"."DETAIL_CODE" (DETAIL_CODE_ID, PROVIDER_MESSAGE,
VETERAN_MESSAGE) VALUES ('DETCODE18', 'Cancelled by VA - Not Eligible for
Community Care', 'Your request for a Community appointment has been
cancelled because you did not meet eligibility requirements.');
```

```

INSERT INTO "VARDB"."DETAIL_CODE" (DETAIL_CODE_ID, PROVIDER_MESSAGE,
VETERAN_MESSAGE) VALUES ('DETCODE19', 'Cancelled by VA - Unable to meet
request', 'Your request for an appointment has been cancelled because we
have been unable to contact you to complete the request.');
```

## Application Context Override

Create the following application context override if deploying into an **Amazon SMTP** environment like the **Demo** environment in EC2. Create the following file under the {WebLogic\_Domain}/haconfig/schedulingManagerService/v2m2px/config (update x to current version i.e v2m2p2) directory. Other environments may need changes to the properties being defined in the below spring bean.



### applicationContext-smtp.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
  <bean id="mailSender" class="org.springframework.mail.javamail.
JavaMailSenderImpl">
    <property name="host" value="smtp.va.gov" />
    <property name="javaMailProperties">
      <props>
        <prop key="mail.smtp.connectiontimeout">5000</prop>
        <prop key="mail.smtp.timeout">5000</prop>
      </props>
    </property>
  </bean>
</beans>
```

## Deployment

**SchedulingManagerService** is a standard VAMF web service that has been developed and tested on WebLogic. After verifying the dependencies and property configuration, install the *Deployable Artifact* from the Release Documentation Page (parent) page on the WebLogic cluster/server.

Deployment of WAR file can be performed either manually through the WebLogic Admin Console or through the following sample script:

```
# Please make sure the paths are correct and the parameters are replaced
with correct values before executing the script on command line

. /opt/oracle/middleware/user_projects/domains/<domain>/bin/setDomainEnv.sh
java weblogic.Deployer -adminurl t3://{adminserverip}:6001 -user {userid} -
password {password} -undeploy -name "SchedulingManagerService-2.0.1"

java weblogic.Deployer -adminurl t3://{adminserverip}:6001 -user {userid} -
password {password} -deploy -targets SchedulingManagerServices -name
"SchedulingManagerService-2.0.1" -source "SchedulingManagerService-
{version with build number}.war"
```

## Proxy Configuration

Use the configurations below to setup Apache proxy. Restart the proxy after making changes to the `/etc/httpd/conf.d/scheduling-manager.ssl` configuration file (e.g., `sudo service httpd restart`).

Verify that the following Include statement present in `/etc/httpd2/conf.d/ssl.conf` file, if not add the entry. If the entry is present, no need to add or modify.

### /etc/httpd/conf.d/ssl.conf

```
<VirtualHost *:443>
...

```

```
    Include conf.d/*.ssl
    ...
</VirtualHost>
```

Please add the following proxy configuration to the apache service to forward traffic to the deployment on WebLogic.

#### **/etc/httpd/conf.d/scheduling-manager.ssl**

```
<Location /SchedulingManagerService/v2>
    SetHandler weblogic-handler
    WebLogicCluster <environment-ip-address:port>
    WLProxySSL ON
</Location>
```

## Validation

Ensure that the WebLogic deployment state is **Active** and that <https://<environment-address>/SchedulingManagerService/v2/> returns **"Welcome to SchedulingManagerService"** with the correct manifest.

## Rollback procedures (v2.2.3 to v2.0.3)

1. Revert the configuration changes that were made by deleting the following folder:  
/opt/applications/**haconfig/SchedulingManagerService/v2m2px** (update x to current version i.e v2m2p2) Undeploy the application from the WebLogic cluster.
2. Install v2.0.3 following the instructions here: [Scheduling Manager Resources 2.0.3 - Installation Instructions](#)