OSCAR McMaster

**Technical Manual**

|  |  |
| --- | --- |
| Version: | 0.1 |
| Released: | August 23, 2013 |
| Author: | *Nick Goupinets* |
| File Path: |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Changes** | **Areas Affected** | **Comments** |
| 0.1 | 23-Aug-2013 | Initial release |  |  |

# Objective

This document describes the deployment and administration process. Deployment takes place at the predefined release windows after the system has been tested and approved by the relevant parties as ready for implementation in the production environment.

System deployment consists of configuration of the system and generation of service keys

# Installation Steps

## *System Overview*

eAAPS implementation follows standard OSCAR design guidelines. The main emphasis was to reduce the impact on OSCAR, while implementing the required functionality. This was achieved by encapsulating all the EAAPS-specific modification to OSCAR into its own package (org.oscarehr.research.eaaps), re-using the OSCAR system components and developing "generic" new functionality that can be used in other parts of the system.

Components for exporting patient medical information using Ontario MD data format

1. oscar.oscarDemographic.pageUtil.DemographicExportHelper
2. oscar.oscarDemographic.pageUtil.DemographicExportHelperCallback

UI components that handle export of the OSCAR patient data to the EAAPS server:

1. EaapsExportAction.java - Struts action
2. EaapsExportForm.java - Corresponding struts form
3. ExportEntry.java - component for encapsulating the patient medication information for the export

Code for secure content transfer to the EAAPS server

1. SshHandler.java

Component for integrating EAAPS functionality into the encounter chart

1. EctDisplayEaapsAction.java - Struts action for handling display of the EAAPs widget on the eChart
2. EaapsServiceClient.java - Service client for carrying out the RESTful call to retrieve the EAAPs status
3. EaapsPatientData.java - class for encapsulating the RESTful service response

Component responsible for building the one-way hash from the patient information

1. EaapsHash.java

Classes that are responsible for importing the resulting action plan (as the PDF document)

1. EaapsHandler.java - Handler class for processing HL7 PDF upload

## *System configuration*

eAAPS configuration is carried out via system settings in OSCAR properties. The available properties are listed below

1. eaaps.enabled - boolean flag for enabling or disabling the research module

e.g.: eaaps.enabled=true

1. eaaps.clinic - Name of the clinic that the demographic record belongs to (used for generating the patient hash)

e.g.: eaaps.clinic=Stonechurch

1. eaaps.knownHosts - Location of the known hosts file to be used for exporting medications. This value is optional

e.g.: eaaps.knownHosts=/usr/root/.ssh/known\_hosts

1. eaaps.user - user name for pushing Ontario MD export files with medications to eAAps server

e.g.: eaaps.user=test

1. eaaps.pass - password for pushing Ontario MD export files with medications to eAAps server

e.g.: eaaps.pass=tEsT;

1. eaaps.host - Server URL for pushing Ontario MD export files with medications to eAAps server

e.g.: eaaps.host=knowledgetranslation.ca

1. eaaps.ws.host - End point for the REST web service for the eAAPs status pull

e.g.: eaaps.ws.host=https://secure.knowledgetranslation.ca:8081/eaaps/v0/

1. eaaps.ws.user - User name for the REST web service for the eAAPs status pull

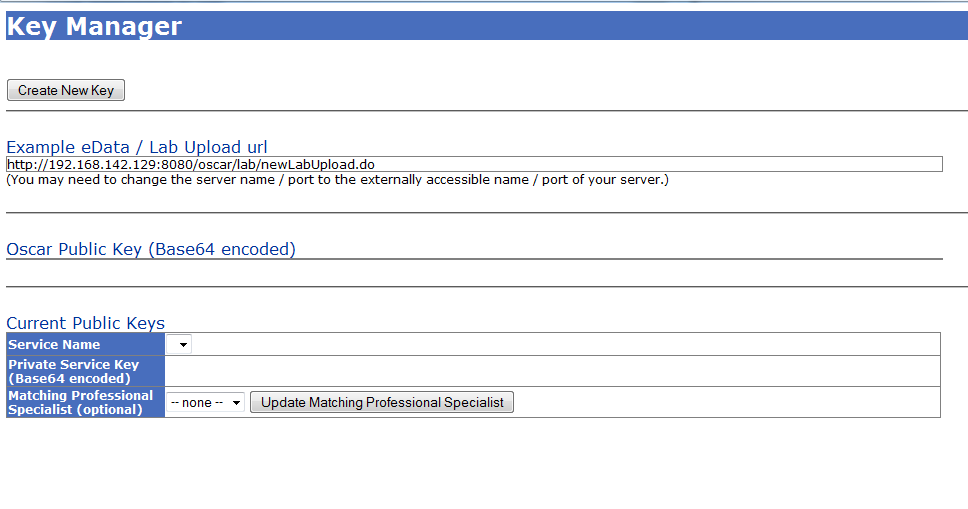
e.g.: eaaps.ws.user=test

1. eaaps.ws.password - Password for the REST web service for the eAAPs status pull

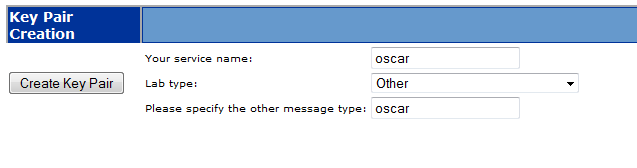
e.g.: eaaps.ws.pass=test

## *Service keys generation*

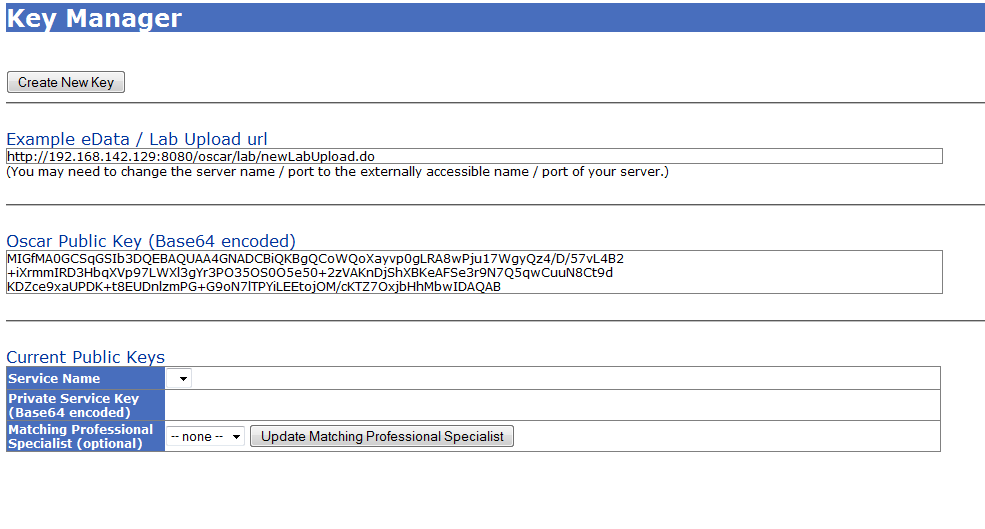
1. Access “Key Manager” via Administraton -> Key Pair Generator



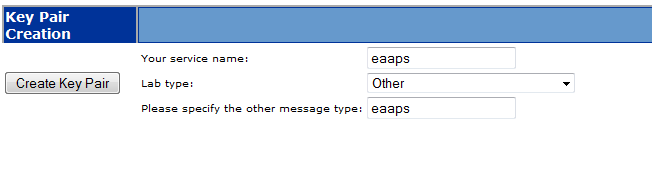
1. Create OSCAR key (if it doesn't exist - i.e. Oscar Public Key is blank, as on the screenshot above. Otherwise go to step 4) by clicking “Create New Key” and enter “oscar” into the service name and message type and click “Create Key Pair”



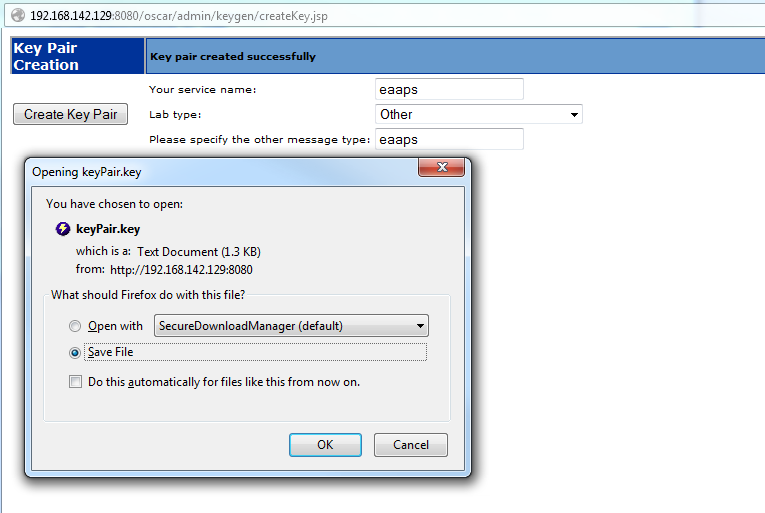
1. Key Manager should now display OSCAR key



1. Create EAAPs service key by clicking “Create New Key” in the Key Manager



1. Enter “eaaps” in service name and lab type and click “Create Key Pair”, saving the key to a secure location



1. Please forward Private Service Key for EAAPs and public OSCAR key to the eAAPs provider who will be responsible for configuration of action plans uploads into your OSCAR installation.

