



Medstrat EMR API

Introduction

Medstrat provides a programming interface for the purpose of integrating its PACS with an EMR. This document describes the division of labor between the EMR and the PACS as well as the details concerning the programming interface to be used as part of an integration. A typical workflow between the EMR and PACS is presented at the end.

Division of Labor

Medstrat provides an application programmer interface (API) for searching and retrieving study data from its PACS. In addition, Medstrat provides a a command-line interface (CLI) in its executables to facilitate opening studies in its viewer based upon a formatted URL.

The EMR will provide a patient-centric graphical user interface (GUI). When a patient context is displayed within that GUI, the EMR will use the API to lookup studies for that patient and display the results as a list within that patient context. When a user selects a (single) study in the list, the EMR will use the CLI to open the study in Medstrat's viewer.

API

The Application Programmers Interface is implemented as an XML-RPC service and is available on port 443 (HTTPS) of the **echoes** server.

```
https://IPAddress/cgi-bin2/echoes-interface.cgi
```

where *IPAddress* above is replaced with the actual IP address of the customer's **echoes** server.



There are a few security aspects to be aware of when using this API:

1. The CGI script is protected by basic authentication. Using HTTPS ensures the password is encrypted in transit. The EMR software should prompt its user for a username and password which should then be stored in an encrypted configuration file in the EMR's possession for future use.
2. Although we require HTTPS, we do not use certificates at this time; however, you should be aware that they may be introduced in future versions of the software and will thus be required for the API to operate correctly should that become the case. We will contact vendors in advance when this comes closer to fruition.
3. The first required parameter of any XML-RPC is the *clientKey*. The *clientKey* is different for every **echoes** server and thus the EMR software should store this value in an encrypted configuration file along with the username and password. The *clientKey*'s value will be provided at the time of installation of the customer's **echoes** server when Medstrat and the EMR vendor have the customer in-common.

Supported API RPCs

To search and retrieve study data, the EMR client will call the following remote procedure:

studyList (*clientKey*, *pname*, *pid*, *sdate*, *mod*, *refphys*, *site*, *status*, *advSearch*)

Get all matching rows for the given parameters. The results will include at most 500 rows (for sake of performance).

The following search parameter have "starts with" semantics and will return all results that start with the parameter value: *pname*, *pid*, *refphys*

Date parameters should be formatted as YYYY-MM-DD or YYYY-MM

The modality *mod* parameter accepts the following values: 'CR', 'DR', 'CT', 'MR', 'SC'

For EMRs, it is suggested that the following always be given as empty string: *site*, *status*

For *advSearch*, all parameters have "exactly equal" semantics beginning with server version 2.4.43 but have "starts with" semantics for all prior versions. The *advSearch* parameter is a hash table (hash map, dictionary, etc) whose accepted (string) keys are the following:



'STUINSUID', 'PNAME', 'PID', 'SDATE', 'MODALITY', 'ACCESSION', 'STIME', 'REFPHYS', 'SDESC', 'BODYAREA', 'SEX', 'DOB', 'INSTITUTION', 'NUMSS'

Parameters:

<i>clientKey</i>	The client key to authorize
<i>pname</i>	The patient name (optional)
<i>pid</i>	The patient ID (optional)
<i>sdate</i>	The study date (optional)
<i>mod</i>	The modality (optional)
<i>refphys</i>	The physician (optional)
<i>site</i>	The site's hostname (optional)
<i>status</i>	The study status (optional)
<i>advSearch</i>	Advanced search dictionary (optional)

Returns:

A list of rows where each row is of the form:

Version 2.4: [patientName, patientSex, patientID, studyDate, studyTime, modality, referringPhysician, siteName, status, numberOfDCMs, numberOfJPGs, studyDescription, dateOfBirth, studyInstanceUID, index, institutionName, bodyArea]

Version 3.0: [patientName, patientSex, patientID, studyDate, studyTime, modality, referringPhysician, siteName, status, numberOfDCMs, numberOfJPGs, studyDescription, dateOfBirth, studyInstanceUID, index, institutionName, bodyArea, accessionNumber, numberOfSnapshots]

Each value in the row will be a non-null string (including values representing integers).



version(*clientKey*)

Return the server's software version (inside a list, for backwards compatibility)

Parameters:

<i>clientKey</i>	The client key to authorize
-------------------------	-----------------------------

Returns:

A list containing the version: [version]

Note that some parameters are optional. In this case, reducing the number of arguments to the RPC will result in a correct procedure call whereby default values will be substituted for the missing parameters. The optional parameters will be marked as “optional” in the parameter description.

Command-Line Interface

A command-line is formatted as:

Windows:

```
echoes.exe FLAGS URL
```

Mac:

```
echoes.app/Contents/MacOS/JavaApplicationStub FLAGS URL
```

Where the FLAGS are the following:

-u=	--user=USERNAME	Username to sign in as
-p=	--pass=PASSWORD	Password to sign in with
-n=	--name=UNIQUEID	Name this instance UNIQUEID



Medstrat Executables

Medstrat executables can be found in the following locations:

Windows 32-bit:

```
C:\Program Files\Medstrat\echoes X.Y.Z\echoes.exe
```

Windows 64-bit:

```
C:\Program Files (x86)\Medstrat\echoes X.Y.Z\echoes.exe
```

Mac:

```
/Applications/Medstrat/echoes-X.Y.Z/echoes.app
```

where X.Y.Z is the version of **echoes**. A list of supported Operating Systems can be found [here](#). Medstrat currently only supports 32-bit Windows executables on 64-bit platforms.

Typical Workflow

A typical workflow between an EMR and Medstrat's PACS is as follows:

1. The user selects a specific patient in the EMR and brings up a patient context in its GUI.
2. The EMR software fetches a list of studies for that patient by using **studyList** such as:

```
studyList('abc', '', '', '', 'CR', '', '', '', {'PID' : '9876'})
```

where the client key is 'abc', the modality is 'CR', and the patient ID is exactly equal to '9876' (assumes server software version 2.4.43).

3. A list of rows (lists) will be returned in the format defined for that server's software version. The EMR will display the resultant rows in its GUI for that patient context. In addition, the



EMR will keep track of which resultant row is associated with which row displayed in the GUI list.

4. The user then selects one of the study rows in the EMR's GUI. At that point, the EMR software uses the *studyInstanceUID* value from the selected study row and calls the CLI as defined below (the path to the application executable has been elided for sake of brevity):

Windows:

```
echoes.exe -u=USER -p=PASS https://IP/study/SIUID
```

Mac:

```
open -a echoes.app -u=USER -p=PASS https://IP/study/SIUID
```

where IP is the same IP address used when calling the **studyList** remote procedure and SIUID is the Study Instance UID from the resultant row associated with the study selected by the user. The echoes executable will parse the Study Instance UID from the shorthand URL and determine the URLs to all images in the study on behalf of the EMR. After the URLs to the images have been determined, the images will be cached and loaded into Medstrat's viewer.

If the EMR is operating in a multi-tenant environment, such as Windows Server 2003, then the EMR should also use the `-n/--name=UNIQUEID` flag where UNIQUEID is an EMR defined unique identifier. This will ensure that each EMR instance has its own unique viewer instance that will not be affected by any other EMR instance.

In addition, the `-n/--name=UNIQUEID` flag allows the EMR to only experience the overhead cost of the viewer start-up once since that viewer instance's process is re-used during subsequent CLI calls with the same UNIQUEID.

References

For more information on XML-RPC, visit www.xmlrpc.com.