Department of Veterans Affairs

Automated Surgical Risk Calculator (ASRC)

System Administrator Guide



April 2015

Version 1.3

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 11/24/2014 | 1.0 | Document Shell | D. Tombs |
| 11/25/2014 | 1.1 | Technical Edit | S. Vetzel |
| 03/26/2015 | 1.2 | Initial Content | D. Tombs |
| 04/13/2015 | 1.3 | Technical Edit | S. Ambrose |

**Note:** The revision history cycle begins once changes or enhancements are requested after the System Administrator Guide has been baselined.

Table of Contents

[1. System Business and Operational Description 1](#_Toc417388370)

[1.1. Logical System Description 1](#_Toc417388371)

[1.2. Software Description 1](#_Toc417388372)

[1.2.1. Background Processes 1](#_Toc417388373)

[1.2.2. Job Schedules 1](#_Toc417388374)

[1.2.3. Dependent Systems 2](#_Toc417388375)

[2. Routine Operations 3](#_Toc417388376)

[2.1. Administrative Procedures 3](#_Toc417388377)

[2.1.1. System Start-up 3](#_Toc417388378)

[2.1.2. System Shut-down 3](#_Toc417388379)

[2.1.3. Back-up & Restore 3](#_Toc417388380)

[2.1.3.1. Back-Up Procedures 3](#_Toc417388381)

[2.1.3.2. Restore Procedures 3](#_Toc417388382)

[2.1.3.3. Back-Up Testing 3](#_Toc417388383)

[2.2. Security/Identity Management 3](#_Toc417388384)

[2.2.1. Identity Management 3](#_Toc417388385)

[2.2.2. Access control 3](#_Toc417388386)

[2.3. System Monitoring, Reporting & Tools 4](#_Toc417388387)

[2.3.1. Monitoring Tools 4](#_Toc417388388)

[2.3.1.1. Monitoring with JConsole 4](#_Toc417388389)

[2.3.1.2. Monitoring via the GlassFish Administration Console 5](#_Toc417388390)

[2.3.2. Availability Monitoring 6](#_Toc417388391)

[2.3.3. Performance/Capacity Monitoring 6](#_Toc417388392)

[2.3.4. Critical Metrics 6](#_Toc417388393)

[2.4. Routine Updates, Extracts and Purges 6](#_Toc417388394)

[2.4.1. Updating the ASRC Software 6](#_Toc417388395)

[2.5. Scheduled Maintenance 7](#_Toc417388396)

[2.6. Capacity Planning 7](#_Toc417388397)

[3. Exception Handling 8](#_Toc417388398)

[3.1. Routine Errors 8](#_Toc417388399)

[3.1.1. Security Errors 8](#_Toc417388400)

[3.1.2. Time-outs 8](#_Toc417388401)

[3.1.3. Concurrency 8](#_Toc417388402)

[3.2. Significant Errors 8](#_Toc417388403)

[3.2.1. Application Error Logs 8](#_Toc417388404)

[3.2.1.1. Primary Application Log 8](#_Toc417388405)

[3.2.1.2. Other Logs 9](#_Toc417388406)

[3.3. Dependent System(s) 9](#_Toc417388407)

[3.4. Troubleshooting 9](#_Toc417388408)

[3.4.1. *HTTP Status 404* 9](#_Toc417388409)

[3.5. System Recovery 9](#_Toc417388410)

[3.5.1. Restart after Non-Scheduled System Interruption 9](#_Toc417388411)

[3.5.2. Restart after Database Restore 10](#_Toc417388412)

[3.5.3. Rollback Procedures 10](#_Toc417388413)

List of Figures

[Figure 1 – Dependent Systems 2](#_Toc417388414)

[Figure 2 - Available Tools for Monitoring ASRC 4](#_Toc417388415)

[Figure 3 - Connecting via Jconsole 5](#_Toc417388416)

# System Business and Operational Description

The Automated Surgical Risk Calculator (ASRC) Tool is used at the time the patient is considered for surgical referral by a primary care provider, and when a surgeon is requesting a surgery. This Tool will support clinical decision-making regarding perioperative risk (includes preoperative, intraoperative, and postoperative).

Providers will verify patient-specific data that is automatically pulled from available data sources, enter remaining fields, and be provided with a real-time individual risk calculation of perioperative surgical mortality based on historic Veterans Affairs Surgical Quality Improvement Program (VASQIP) data and current VASQIP risk-adjusted models that are specialty-specific. The data entered and the calculated results will be available for viewing in the Computerized Patient Record System (CPRS) as a progress note. The data will also transfer and store as discrete fields in Veterans Health Systems and Technology Architecture (VistA) and a Structured Query Language (SQL) database for use by the National Surgery Office (NSO).

## Logical System Description

The system consists of only one logical component, the ASRC Web Application. This component interfaces with other systems such as VistA.

There is an associated VistA Surgery patch to support ASRC but VistA Surgery is not considered a component of the system for the purposes of this document.

See the ASRC Technical Manual for more information.

## Software Description

The ASRC Web Application is a Java Enterprise application running on Glassfish 3, a Java Enterprise Edition (Java EE) Application Server. It uses MySQL as the backend relational database. The target host operating system is Windows Server 2012 R2.

See the ASRC Technical Manual for more information.

### Background Processes

The only background processes that run as part of the system are provided by the off-the-shelf software, namely:

* The Glassfish Server
* The MySQL Server

### Job Schedules

The system has no associated periodic jobs.

### Dependent Systems

The systems on which the ASRC system depends are depicted below:



Figure – Dependent Systems

VistA is required for normal operation of ASRC.

The *National ASRC Results Database* could be considered part of the ASRC system itself, but other users access the database so it is depicted separately.

# Routine Operations

To be determined once more of the system is designed and implemented.

## Administrative Procedures

### System Start-up

No manual action must be taken to start ASRC beyond simply booting the host web server. All services are configured to start automatically.

### System Shut-down

To shut down the ASRC system, simply shut down the host web server.[[1]](#footnote-1) Since any user operations in progress will be lost, ensure that no users are accessing the system when it is shut down.

### Back-up & Restore

To be determined once more of the system is designed and implemented.

#### Back-Up Procedures

To be determined once more of the system is designed and implemented.

#### Restore Procedures

To be determined once more of the system is designed and implemented.

#### Back-Up Testing

To be determined once more of the system is designed and implemented.

## Security/Identity Management

This section is pending design and implementation of user and administrator authentication and authorization.

### Identity Management

This section is pending design and implementation of user and administrator authentication and authorization.

### Access control

This section is pending design and implementation of user and administrator authentication and authorization.

## System Monitoring, Reporting & Tools

This section describes the high-level approach to system monitoring.

The following diagram depicts the tools available for monitoring the system.



Figure - Available Tools for Monitoring ASRC

### Monitoring Tools

The two best methods of monitoring the ASRC system are JConsole (provided with the Java Development Kit (JDK)) and the Glassfish Administration Console.

#### Monitoring with JConsole

The JDK includes a monitoring tool called JConsole, which connects to a running Java Virtual Machine (JVM) via Java Management Extensions (JMX). JConsole provides graphs of the JVM memory and thread statistics as well as a low-level view of various Management Beans (MBeans) provided by the JVM and Glassfish.

To monitor ASRC via JConsole:

1. Open a Remote Desktop connection to the target ASRC Web Server.
2. Run %JAVA\_HOME%\bin\jconsole.exe.
3. Find the line in server.log (see Primary Application Log) the prints the JMX URL, e.g.:  
     
   [#|2015‑03‑26T17:35:41.565+0000|INFO|glassfish3.1.2|javax.enterprise.system.jmx.org.glassfish.admin.mbeanserver|\_ThreadID=65;\_ThreadName=Thread-2;|JMX005: JMXStartupService had Started JMXConnector on JMXService URL **service:jmx:rmi://10.146.174.8:8686/jndi/rmi://10.146.174.8:8686/jmxrmi**|#]
4. Enter the JMX URL (in bold above) into the *Remote Process* text box.
5. Enter *admin* into the *Username* box.
6. Enter the configured password into the *Password* box. (This was configured during server installation, see the ASRC Installation Guide.)
7. Click *Connect*.

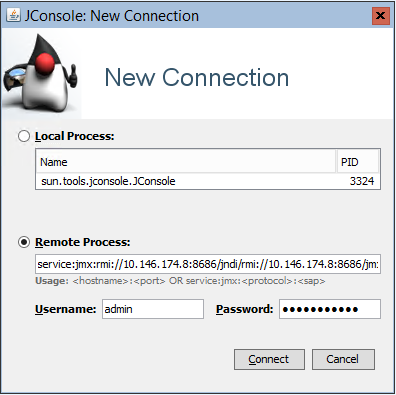


Figure - Connecting via Jconsole

A discussion of the various data available via JConsole is out of the scope of this document. For more information, see Oracle’s Using JConsole[[2]](#footnote-2).

#### Monitoring via the GlassFish Administration Console

The GlassFish Server Administration Console provides a browser interface for configuring, administering, and monitoring GlassFish Server.

To access the console:

1. Open a Remote Desktop connection to the target ASRC Web Server.
2. Open Internet Explorer and navigate to <http://localhost:4848/>. (Although this is a web interface, it is only accessible from localhost.)
3. Enter credentials.
   1. *User Name*: admin
   2. *Password*: the password configured during installation.
4. Click *Login*.
5. On the left navigation bar, click *Server (Admin Server)*.
6. On the top tab bar, click *Monitor*.
7. Various monitoring data are available on the sub-tabs.

Note that, although GlassFish provides many monitored points, not all monitoring is enabled out-of-the-box. See the GlassFish Administration Guide for information on enabling monitored points.

### Availability Monitoring

To be determined after more of the system has been designed and implemented.

### Performance/Capacity Monitoring

To be determined after more of the system has been designed and implemented.

### Critical Metrics

To be determined after more of the system has been designed and implemented.

## Routine Updates, Extracts and Purges

The only routine maintenance defined for the ASRC system is a software update. No routine extracts or purges are required.

### Updating the ASRC Software

Use the following process to update the ASRC software deployed to a server.

1. Ensure no users are using the system.
2. Download the desired release zip file from the project’s GitHub repository (<https://github.com/VHAINNOVATIONS/ASRCM/releases/>).
3. Extract the zip file to any directory on the server accessible by your user. This unzipped directory will hereafter be referred to as the *Installation Files Directory*.
4. Build the srcalc.war application archive (see the ASRC Developer Guide) and transfer it to the server. Place it into the install directory in the *Installation Files Directory*.
5. Update the ASRC Database. (**Warning:** at time of writing, all data in the database is static. There is no transactional data. The below process therefore updates the database by deleting and re-creating it. This process must be changed when the application starts making changes to the database to avoid data loss.)
   1. Open a Command Prompt and change directory to the *Installation Files Directory*.
   2. Execute: cd install
   3. Execute: config.bat
   4. Execute: echo drop database srcalc; | %MYSQL% -u root
   5. Execute: create\_database.bat
   6. The ASRC database is now updated.
6. Using the same command prompt, execute: deploy.bat
7. Launch the application from CPRS as usual to smoke-test the deployment.

## Scheduled Maintenance

As ASRC is currently an innovation project and not in production, no scheduled maintenance is defined. A maintenance schedule should be defined before the system is in production.

## Capacity Planning

As ASRC is currently an innovation project and not in production, no capacity planning process is defined. A process should be defined before the system is in production.

# Exception Handling

This section is to be determined nearer to the end of the innovation project.

## Routine Errors

Like most systems, ASRC may generate a small set of errors that may be considered routine in the sense that they have minimal impact on the user and do not compromise the operational state of the system. Most of the errors are transient in nature and only require the user to retry an operation. The following subsections describe these errors, their causes, and what, if any, response an operator needs to take.

While the occasional occurrence of these errors may be routine, getting a large number of an individual error over a short period of time is an indication of a more serious problem. In that case the error needs to be treated as an exceptional condition.

### Security Errors

This section is to be determined nearer to the end of the innovation project.

### Time-outs

This section is to be determined nearer to the end of the innovation project.

### Concurrency

This section is to be determined nearer to the end of the innovation project.

## Significant Errors

Significant errors can be defined as errors or conditions that affect the system stability, availability, performance, or otherwise make the system unavailable to its user base. The following subsections contain information to aid administrators, operators, and other support personnel in the resolution of errors, conditions, or other issues.

### Application Error Logs

This section describes the applications error logging functionality, lists the locations of where they are stored and what, if any, special tools are needed to view the log entries.

#### Primary Application Log

The primary log file of the ASRC system is the Glassfish server log:

C:\asrc\glassfish3\glassfish\domains\domain1\logs\server.log

Glassfish automatically rotates this log file when it reaches 2MB. Rotated files are named server.log\_*date*, which *date* is the date and time the log file was rotated. Ten rotated log files are retained.

All of these files are plaintext with newline-delimited log entries. Each entry has an associated level (e.g., SEVERE or INFO), and certain log entries may be suppressed if their associated levels are below the configured level. For example, if the configured log level is INFO, then a FINE-level log entry will be omitted from the log file. The configured level can be changed while the server is running using the GlassFish Administration Console. (See Monitoring via the GlassFish Administration Console above.)

See the *GlassFish Administration Guide*, chapter 7, for more information on the log files’ format and rotation.

#### Other Logs

MySQL and system services log to the Windows Event Log. These logs may be viewed with the Windows Event Viewer. MySQL logs to the Application log, other system services log to various other logs.

The ASRC system does not send logs or any kind of alarm messages to external systems.

## Dependent System(s)

The only system on with ASRC depends for normal operation is VistA. If VistA is inaccessible or behaving abnormally, the users may be able to launch the application but will receive an error mentioning the VistA interface. Note that one deployment of ASRC may communicate with multiple VistA instances, so ASRC may operate normally for some users but not for others.

To resolve VistA communication errors, contact the support team for the appropriate VistA instance.

## Troubleshooting

This section contains helpful information on troubleshooting the system that has been learned as part of the development and testing process, or from the operation of similar systems. The information is grouped into sub-sections based on visible symptoms.

### *HTTP Status 404*

If users receive an *HTTP Status 404* error upon launching the application, check the server log (see Primary Application Log) for any deployment errors.

The most common error is that the MySQL database server is not running. In this case, the server log will contain an exception such as:

org.hibernate.HibernateException: Connection cannot be null when 'hibernate.dialect' not set

To resolve, start the MySQL database server and restart the GlassFish server.

## System Recovery

The following subsections define the process and procedures necessary to restore the system to a fully operational state after a service interruption. Each of the subsections starts at a specific system state and ends up with a fully operational system.

### Restart after Non-Scheduled System Interruption

If the application crashes, simply restart the host web server by following the procedures in System Shut-down and System Start-up.

### Restart after Database Restore

To be determined after completing the backup and restore sections.

### Rollback Procedures

The MySQL Database Server automatically performs rollback procedures after a server crash and should bring the database to a consistent state. See Section 14.16.1 of the MySQL Reference Manual[[3]](#footnote-3) for more information on the crash recovery process.

1. Both GlassFish and MySQL are configured as Windows Services, so they shut down cleanly when the operating system shuts down. Shutdown order is not guaranteed, thus the database may shut down before the application: hence the importance of ensuring no users are currently accessing the system. [↑](#footnote-ref-1)
2. <http://docs.oracle.com/javase/7/docs/technotes/guides/management/jconsole.html> [↑](#footnote-ref-2)
3. <http://dev.mysql.com/doc/refman/5.6/en/innodb-recovery.html> [↑](#footnote-ref-3)