Perceptive Reach

Integrated Reach Database System

(IRDS)

System Administration Manual



Department of Veterans Affairs

**July 2015**

**Version 0.3**

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 07/14/2015 | 0.3 | Document updates | Andal FeQuiere |
| 06/12/2015 | 0.2 | Document updates | Andal FeQuiere and Bill Balshem |
| 05/10/2015 | 0.1 | Initial Version | Matthew Robinson and Radina Ivanova |

Artifact Rationale

Table of Contents

[1. System Business and Operational Description 1](#_Toc420661212)

[1.1. Operational Priority and Service Level 1](#_Toc420661213)

[1.2. Logical System Description 2](#_Toc420661214)

[1.3. Physical System Description 3](#_Toc420661215)

[1.4. Software Description 4](#_Toc420661216)

[1.4.1. Background Processes 5](#_Toc420661217)

[1.4.2. Job Schedules 5](#_Toc420661218)

[1.4.3. Dependent Systems 5](#_Toc420661219)

[2. Routine Operations 6](#_Toc420661220)

[2.1. Administrative Procedures 6](#_Toc420661221)

[2.1.1. System Start-up 6](#_Toc420661222)

[2.1.2. System Shut-down 6](#_Toc420661223)

[2.1.3. Back-up & Restore 7](#_Toc420661224)

[2.1.3.1. Back-Up Procedures 7](#_Toc420661225)

[2.1.3.2. Restore Procedures 9](#_Toc420661226)

[2.1.3.3. Back-Up Testing 9](#_Toc420661227)

[2.1.3.4. Storage and Rotation 10](#_Toc420661228)

[2.2. Security / Identity Management 10](#_Toc420661229)

[2.2.1. Identity Management 11](#_Toc420661230)

[2.3. User Notifications 12](#_Toc420661231)

[2.4. System Monitoring, Reporting & Tools 12](#_Toc420661232)

[2.4.1. Availability Monitoring 12](#_Toc420661233)

[2.4.2. Performance/Capacity Monitoring 12](#_Toc420661234)

[2.4.3. Critical Metrics 13](#_Toc420661235)

[2.5. Routine Updates, Extracts and Purges 14](#_Toc420661236)

[2.6. Scheduled Maintenance 15](#_Toc420661237)

[2.7. Capacity Planning 15](#_Toc420661238)

[3. Exception Handling 16](#_Toc420661239)

[3.1. Routine Errors 16](#_Toc420661240)

[3.1.1. Security Errors 16](#_Toc420661241)

[3.1.2. Time-outs 17](#_Toc420661242)

[3.1.3. Concurrency 18](#_Toc420661243)

[3.2. Significant Errors 18](#_Toc420661244)

[3.2.1. Application Error Logs 18](#_Toc420661245)

[3.2.2. Application Error Codes and Descriptions 19](#_Toc420661246)

[3.2.3. Infrastructure Errors 20](#_Toc420661247)

[3.2.3.1. Database 21](#_Toc420661248)

[3.2.3.2. Web Server/Application Server 21](#_Toc420661249)

[3.2.3.3. Network 21](#_Toc420661250)

[3.2.3.4. Authentication & Authorization 21](#_Toc420661251)

[3.3. Dependent System(s) 22](#_Toc420661252)

[3.4. Troubleshooting 22](#_Toc420661253)

[3.5. System Recovery 26](#_Toc420661254)

[3.5.1. Restart after Non-Scheduled System Interruption 26](#_Toc420661255)

[3.5.2. Restart after Database Restore 27](#_Toc420661256)

[3.5.3. Back Out Procedures 27](#_Toc420661257)

[4. Operations & Maintenance System Support 29](#_Toc420661258)

[4.1. Support Structure 29](#_Toc420661259)

[4.1.1. Support Hierarchy 29](#_Toc420661260)

[4.1.2. Division of Responsibilities 29](#_Toc420661261)

[4.2. Support Procedures 29](#_Toc420661262)

# System Business and Operational Description

This System Administration Manual (SAM) is developed for the Perceptive Reach Integrated Reach Database System (IRDS) project for the Department of Veterans Affairs (VA). VA is seeking to expand suicide prevention to include upstream approaches designed to reduce initiation or escalation of a suicide risk factor. Upstream suicide interventions target individuals or groups who exhibit biological, psychological, or social risk factors that are more prominent among high-risk groups than among the larger population. Understanding the unique needs of our nation’s Veterans and the military culture as it relates to stigma and mental health is important for early intervention. The goal of the IRDS innovation is to promote the general health of the Veteran population and effectively intervene in issues before they escalate into crisis.

The IRDS solution will harness the power of large and diverse data stores to aggregate, analyze and identify risk onset as well as reveal previously unidentified at-risk individuals and populations as a holistic and integrated approach. The IRDS innovation will serve to bolster the three major components of Veterans Health Administration’s (VHA) Strategic Plan for Suicide Prevention: surveillance, risk and protective factors, and prevention interventions. The IRDS innovation will target antecedent events specific to Veteran populations prior to the onset of risk to mitigate the development of risk.

The primary stakeholders for this project are:

* Dr. Robert Bossarte
* Dr. Caitlin Thompson
* Clint Latimer

.

## Operational Priority and Service Level

IRDS’s overall importance to VA lies within its ability to augment current suicide prevention programs with supporting data for proactive, “upstream” suicide prevention approaches. To do this, the application applies sophisticated predictive analytics frameworks to Veteran health data. The application of these frameworks produces stratified lists of Veterans who are potentially at an increased risk for suicidal ideation or attempt. Clinical providers, such as VA Suicide Prevention Coordinators (SPCs) can then use this information to reach out to Veterans. The customers served are the Veterans who consume VHA-provided services, VHA clinicians, and supervisors. In addition, parties interested in research and reporting based on the data held in the IRDS will be granted special access to perform these tasks.

IRDS is considered *important*, but not *critical* for patient care, in most cases. If the system is down or otherwise unavailable, patients’ safety will not be immediately compromised and standard VA programming for suicide prevention will be available as a contingency. However, the system’s unique ability to identify and report Veterans at an increased risk for suicidal ideation or other behavioral health issues in real time implies that the service should be prioritized above other non-critical systems.

## Logical System Description

The Perceptive Reach program, including components of the IRDS, are as follows:

* Reach Database – a Structured Query Language (SQL) database storing data used for analytic input
* Data Analytics Platform – an integrated collection of analytic tools
* Risk Model – the predictive model(s) used to identify high-risk Veterans
* Dashboard – a multi-view information portal displaying results from the analytic platform and risk model
* Direct Messaging – a secure messaging solution to notify outreach and intervention resources/clinicians about high-risk Veterans



Figure 1: IRDS System Overview

The primary ingress interfaces will use SQL Server Integration Services (SSIS) to retrieve data from the designated data sources such as the SDR. The SSIS package will be configured to pull data from the designated data source at a specified interval (daily/weekly/monthly/annually). The primary egress interface will be the Direct Messaging interface used to notify VA Suicide Prevention Coordinators (SPC) and other frontline care coordinators and clinicians. The IRDS will leverage the VLER Direct messaging service utilizing a RESTful interface to transmit message content. Another interface utilized by VA Staff and clinical providers such as SPCs will be the IRDS Surveillance Dashboard which will provide browser based visualizations of critical data to identify at-risk Veterans.



Figure 3: IRDS Interface Overview

Figure 4: Logic Data Integration

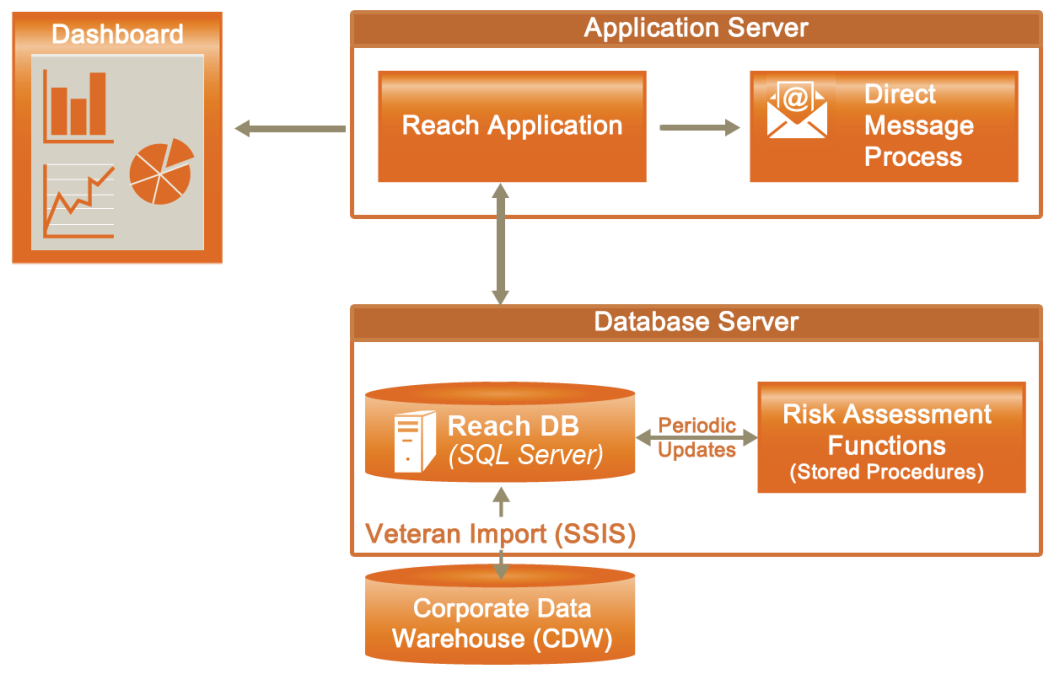
## Physical System Description

:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Type** | **OS** | **Processor** | **Memory** | **Storage** | **Users** | **Applications** |
| Cloud (Database Server) | MS Windows Server 2012 64-bit | Intel Xeon CPU E5-2670, 2.6GHZ | 16 GB | 60 GB | 16 | MS SQL Server 2012 Enterprise Edition |
| Cloud (Application Server) | MS Windows Server 2012 64-bit | Intel Xeon CPU E5-2670, 2.6GHZ | 16 GB | 60 GB | 16 | IIS 7.0 |

Table 1: System Hardware

## Software Description

The system centers on the Reach Database as depicted in **Figure** 6. Veteran data flows in from VA sources, initially CDW, and is processed through a Risk Assessment which categorizes each Veteran at a risk level. Based on requirements, the IRDS Dashboard and Direct Message notification process will interact with veteran data to provide support through visualizations and notifications to VA SPCs.

**Figure 6: Application Design**

A full list of the software used in the system is described below:

| Name | Technologies | License |
| --- | --- | --- |
| Dashboard | HTML5, CSS3, JavaScript, JQuery, Angular JS | Open source |
| Direct Message Assembler | VLER Direct API | VA |
| Data Analytics Platform | KNIME | Open source |
| Reach DB | SQL Server 2012 | Commercial (provided) |
|  |  |  |
| Web Server | IIS, Windows Server 2012 | Commercial (provided) |

Table 3: Software Used in IRDS

All application software is open source or provided by the VA. The operating system is Windows due to the dependency on using SQL Server.

### Background Processes

The application background processes are as follows:

* InetMgr.exe: The IIS application which hosts the web application
* sqlservr.exe: The server daemon for the SQL Server database
* node.exe: The JavaScript runtime technology to host our server side application

**Note:** SQL Server is configured to run as a Windows service so that it starts automatically with Windows.

### Job Schedules

As the system gets close to its production, the jobs that will be run and their scheduling will be defined.

### Dependent Systems

| **Name** | **Description** | **Interface Name** | **Interface System** |
| --- | --- | --- | --- |
| Direct Message Assembler | To facilitate intervention through outreach programs, the IRDS system will create and transmit notification messages via Direct Messaging to VA designated and authorized intervention service providers. | SQL Query, HTTPS | Reach DB, VLER Direct |
| Corporate Data Warehouse (CDW) | Warehouses multiple VBA and VHA data sources in SQL Format. | SSIS, SQL Query | Reach DB |

**Table 4: Dependent Systems**

# Routine Operations

There are two routine operations that must be performed on the system: user management and backing up the database. User management is performed within the administrative dashboard and consists of adding, editing, and removing users, and is covered in Section 2.2, Security/Identity Management. Backing up the database is performed at the operating system level and is covered in Section 2.1.3 Back-up &Restore.

| Operation | Role | Section |
| --- | --- | --- |
| User management | Program technical administrator | 2.2 |
| Database backup | System administrator | 2.1.3 |

Table 5: Routine Operations

## Administrative Procedures

### System Start-up

The system does not require any regular manual start-up procedures. The database and application servers are both implemented as Windows services that automatically start with Windows, and it is unlikely that either service will fail under normal conditions. The services are listed below:

| Service | Name | Display Name |
| --- | --- | --- |
| SQL Server (MSSQLSERVER) | MSSQLSERVER | MSSQLSERVER |
| Windows Process Activation Service | WAS | WAS |
| World Wide Web Publishing Service | W3SVC | W3SVC |

Table 6: System Services

In the event that either does not start, or shuts down prematurely, they can be manually started via the Windows services snap-in. However, before starting either service, consult the Windows event viewer and individual service logs for information about the error. See Section 3, Exception Handling for more details about error handling and logs.

### System Shut-down

The system can be shut down by shutting down the two system processes from the Services snap-in. The services are listed in Section 2.1.1, System Start-up.

### Back-up & Restore

Database backup and restoration is performed and managed by the AITC staff assigned to the IRDS system. The IRDS development team will communicate to the AITC staff which database on the IRDS database server need to be backed up. For those databases, AITC will perform hot backups directly to tape on a nightly basis.

The IRDS development team has the option create a backup to disk on the data (E:) drive when necessary, such as when updates are being applied to production database. In the event that an issue occurs. The database can quickly be restored from backup saved to disk.

#### Storage and Rotation

The VA system administrator should copy/move database backups to remote storage for safekeeping according to IT guidelines. The IRDS server provides RAIDed storage and a rolling backup system for safeguarding backups locally, but relocating backups to SAN or other storage per VA IT guidelines provides additional safety and redundancy. External/redundant storage by IT is beyond the scope of this document.

## Security / Identity Management

The IRDS security architecture consists of components that perform authentication and authorization of VA/Non-VA staff. These components include the following:

* Network: Encrypted and secured communication between the Web application and the server over HTTPS
* Web application: Authenticated and authorized access to features
* CDW: Authenticated and authorized access to pull some basic data and upload data

The following attributes describe the IRDS architecture as related to security:

* The IRDS system resides in the SD data center and consists of a web application, web services, and a database
* VA/Non-VA Users access authorized portions of the web application from VA facilities over VA WIFI/VPM using the users network account credentials
* IRDS reads limited patient identification and demographics data from CDW, and writes results to the Reach DB
* IRDS integrates with CDW via SSIS entirely on the VA network

For more information on IRDS security, see the System Security Plan.

### Identity Management

Users are added, modified, or deactivated through the Users table within the SQL Server Reach Database. In the Users table interface, the technical admin user has the ability to create users, assign their access (add them to program locations), and deactivate them. Adding a user consists of using the Userstable to fill in the new user’s username, user role, user state location, first name, last name, home facility, domain, and active/inactive status. Modifying a user consists of using the User table to modify values. Deactivating consists of changing the user’s status to *inactive*.

**Note:** Users are not deleted in IRDS; they are simply inactivated and they can be reactivated in the future as needed.

|  |  |  |
| --- | --- | --- |
| Activity | Location | Interface |
| Add user | SQL Server Reach Database | User Table |
| Edit user | SQL Server Reach Database | User Table |
| Deactivate user | SQL Server Reach Database | User Status field |

Table 10: Identity Management Functions

For more information on user management, see the User Guide.

## User Notifications

The user community will be notified of any scheduled changes via email distribution lists. It is recommended that separate mailing lists should be established for users, program administrators, and support staff.

## System Monitoring, Reporting & Tools

System monitoring should be performed and managed by the AITC staff assigned to the IRDS system. System monitoring tools should be established for operating system CPU, memory, disk space, and the IIS and SQL Server processes.

### Availability Monitoring

The IRDS availability monitoring requirements will be determined by the AITC staff assigned to the IRDS system. The main components that will be monitored are the web-based Dashboard application (IIS), database (SQL Server), and CDW connectivity.

| Concern | Test |
| --- | --- |
| Web-based Dashboard application | Application: OK |
| Database connectivity | Database: OK |
| CDW connectivity | CDW: OK |

Table 11: Application Status Checks

Loading the status screen checks the application status in general, as well as the database and CDW connectivity. In addition to automated monitoring, this screen can be checked manually to determine the status of the system after a deployment or patch, or during troubleshooting.

### Performance/Capacity Monitoring

IRDS performance and capacity management consists of two concepts: verifying system performance through page loads, and verifying capacity through disk and network analysis.

There are three stated KPIs for IRDS: initial page load <= 20s and subsequent page load <= 5s. These KPIs are summarized below:

| Action | Threshold | Verification |
| --- | --- | --- |
| Initial page load | 20 seconds | Manual |
| Subsequent page load | 5 seconds | Manual |

Table 12: Performance Thresholds

The responsibilities of performance and capacity monitoring belong to AITC, since the IRDS solution will be hosted within AITC servers. For more details on AITC’s procedures and monitoring tools please contact AITC.

### Critical Metrics

The critical metric for IRDS is whether 100 concurrent users can simultaneously use the system nationally. The downstream implication is a possible delay in identifying or seeking treatment for some Veterans. The critical metric is summarized below:

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Threshold | Upstream implications | Downstream implications |
| Concurrent users nationally | 100 | Delay or errors accessing the surveillance Dashboard application | Delayed identification of at-risk Veterans |

Table 14: Critical Metrics for IRDS

The system’s current or historical support for concurrent users/site can be assessed by exporting the log files to VA’s enterprise log analysis service. The logs have an industry standard structure that will be recognized without custom parsing by most commercial or open source log parsing tools. Adherence can be determined by comparing page requests times and error counts against concurrent logins.

## Routine Updates, Extracts and Purges

Updates, extracts, and purges are performed for IRDS per VA guidelines and as requested by program administrators. These activities are summarized below:

| Activity | Periodicity | Responsible party |
| --- | --- | --- |
| Updates | As needed | DBA |

Table 15: Routine Data Activities

Updates consist of inserting or updating data in the database can be performed as needed. Updates should be scripted with sufficient error handling and rollback logic to handle expected and unexpected errors during execution while protecting data integrity. Data changes require expertise in SQL and the IRDS schema (see project schema document). Updates should be performed by qualified DBAs as requested by IRDS program coordinators.

## Scheduled Maintenance

Scheduled maintenance will be performed as authorized by program administrators. Currently, there is no schedule for maintenance.

## Capacity Planning

Capacity planning should be performed by VASD IT in cooperation with IRDS program administrators. Currently, there is no schedule or requirements for capacity planning.

# Exception Handling

Runtime errors in IRDS are typically related to configuration, connectivity, or data issues. Errors related to connecting to the IRDS database, configuration, and bad or unmatched data can be resolved locally by the system administrator. Other kinds of errors, such as problems connecting to the CDW can be resolved through cooperating with external teams. The types of errors are summarized below:

| Type | Examples |
| --- | --- |
| Locally resolvable | Unmatched records, bad data, DB connectivity |
| Externally resolvable | Network or CDW issues |
| Unresolvable | Errors due to bugs |

Table 16: Types of Errors

**Note:** Some errors, such as those due to unidentified bugs, require application source code changes and cannot be changed by the system administrator.

## Routine Errors

Like most systems, IRDS 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, observing 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

Security errors in IRDS consist of authentication or authorization issues related to accounts. For example, if a staff user attempts to logon to IRDS with invalid credentials three contiguous times, the system will display an error message directing the user to contact the system administrator.

Likewise, if a staff user is already authenticated but attempts an unauthorized action in the application, the system will display an error message

The following table displays the known security error types, descriptions, and resolutions:

| Type | Description | Resolution |
| --- | --- | --- |
| Staff authentication failure | When a staff user does not have a valid account or does not use a valid password, the user will be unable to login and an error message will display instructing the user to enter valid username and password credentials. | Ensure user is active in IRDS and provide appropriate username and password. |
| Staff account locked | When a staff user fails to successfully login the dashboard after 3 attempts, their account will be locked for a period of 24 hours or can be resolved by a system administrator. | Set *LoginAttemptCount* field in Users Table within the SQL Server Reach DB to a value of *0*. |
| Staff account inactive | When a staff user fails to successfully login the dashboard after 30 days, their account will be deactivated and can only be resolved by a system administrator. | Set *isActive* field in Users Table within the SQL Server Reach DB to a value of *True*. |
| Staff account max sessions reached | When a staff user has 3/1 active sessions open, has a role of system admin/non-system admin, and attempts to open an additional session, they will be unable to login and will receive instructions to resolve the issue. | Ensure user successfully logout of an active session to free up available sessions. |

Table 17: Security Errors in IRDS

### Time-outs

In IRDS, timeouts can occur between the client and the server, and between the server and CDW. Timeouts can be due to capacity issues with regard to the IRDS server, the CDW server, or the network fabric in between. It is expected that most timeouts will be due to capacity or contention issues caused by the browser communicating with the server over VA network, no on the server itself or between the server and CDW. When any part of the system times out, the application displays a user-friendly error message indicating that the user should talk to the clerk. The following table summarizes the types of possible timeouts:

| Type | Incidence | Response |
| --- | --- | --- |
| Timeout connecting to CDW | Unknown | Repeat attempt or file support ticket with CDW or NOSS group. |
| Timeout loading data to IRDS | Unknown | Try again, troubleshoot server, or file support ticket with NOSS or AITC |

Table 18: Possible IRDS Timeouts

Most timeouts will be transient in nature, and resolve after the network or server contention abates. However, timeouts can also be investigated and submitted to the appropriate support groups. Some timeouts between the server and client can be logged, timeouts on the server itself, and timeouts between the server and CDW are logged on the server. This allows the system administrator to investigate individual timeout issues as well as use system tools or external tools in order to investigate patterns of timeouts.

For example, the system administrator can see the most-recent errors by tailing the log in PowerShell:

gc d:\data\logs\logFileName.log –tail 100

Likewise, the admin can search all log files for timeout errors:

dir d:\data\logs\\*.log | select-string “connection refused”

Timeouts tend to be sporadic, based on transient network or server conditions. However, the system administrator can analyze the logs in VA’s enterprise log analysis utility for greater insight into trends.

### Concurrency

Concurrent updates can lead to unpredictable errors in any system, including IRDS. However, due to the nature of IRDS, concurrency issues are very unlikely to occur. If they did occur, they would be related to very rare events like multiple staff attempting to update a Veteran’s outreach status in the application. In the case of concurrent updates to Veteran data, the system will note if one user is updating an old version of the data and prompt the user to view the updated record and possibly try again.

The table below summarizes the type of possible concurrency issues that could occur:

| Activity | Incidence | Response |
| --- | --- | --- |
| Simultaneous updates to Veteran outreach status in the Dashboard | Very rare | System will catch and log error, then prompt user to (optionally) try again. |

Table 19: Possible Concurrency Issues

## Significant Errors

Significant errors can be defined as errors or conditions that affect the system’s 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

IRDS logs are currently limited to SQL Server Database logs.

Logging for each component is configurable at the component level. The log configuration files are stored in the following folders:

* SQL Server: D:\Program Files\Microsoft SQL Server\MSSQL14\_..\MSSQL\Log\log.ini

Logging is configured by the system administrator. Sensible defaults are supplied along with the application, allowing adequate log coverage for troubleshooting without affecting performance or taking up excessive disk space. The following table outlines the key logging attributes:

| Type | Location | Max size | Growth rate | Rotation | Retention |
| --- | --- | --- | --- | --- | --- |
| SQL logs | D:\ Program Files\Microsoft SQL Server\MSSQL14\_..\MSSQL\Log | As configured | Varies | Daily (suggested) | 10 days (suggested) |

Table 20: IRDS Logging

**Note:** These values should be adjusted by the system administrator based on VA guidelines instead of kept at their default levels.

Querying and analyzing the log files is simple because they are text files and use the industry-standard log conventions (INFO, WARN, ERROR, etc.). PowerShell or other Windows shell utilities can be used to query the files. Some examples are listed in the following table:

| Activity | Example |
| --- | --- |
| Find all errors | select-string "ERROR|SEVERE" \*.log |
| Find warnings in last 100 lines of a file | gc \*.log -tail 100 | select-string WARN |

Table 21: Example Log Queries

For more detailed log file analysis, the system administrator can import the log files with VA’s enterprise log analysis tool.

### Application Error Codes and Descriptions

IRDS does not currently use error codes; rather, it defines custom exception classes that can be used for structured exception handling. These classes can be reused across a family of issues. The table below describes the existing custom application types and descriptions:

| Type | Description |
| --- | --- |
| BadPasswordException | This password is not correct. |
| BadUseridException | User not registered/Invalid Username |
| InactiveUserException | Account is disabled due to inactivity, Contact system admin |
| LockedUserException | Account locked, Please contact system admin |
| MaxSessionsReachedException | Max number of sessions reached, Please log out from your active sessions or wait for 30 seconds and try again |

Table 22: Existing Custom Application Types and Descriptions

### Infrastructure Errors

IRDS relies on various infrastructure components and must handle temporary failures in those components when they occur.

#### Database

IRDS can experience errors connecting to the database or performing data operations. Because the database currently resides on the same server as the application, the most likely cause of database connectivity failures is unhandled exceptions around database connections. Database connection errors can be found in the logs by querying for “connection” and orphaned connections can be queried and forced close via SQL Server commands. For more information on querying and force-closing orphaned connections, see the SQL Server online manual: <https://support.microsoft.com/en-us/kb/137983> .

The application can experience errors performing data operations as well. This includes errors querying, inserting, updating, or deleting data. When these types of database errors occur, the application will catch the exception and log it. If the error is something that the user can fix by trying again, the application will display a message to the user; otherwise, the application will handle the error itself and may direct the user to a user-friendly error page based on the severity of the error.

#### Web Server/Application Server

Node JS automatically logs all errors to the *stderr* and *stdout* files, although the system administrator can configure the logging per VA guidelines. However, leveraging IIS allows for easier configurability and additional options for logging capabilities. Errors are denoted in the logs by severity (e.g., “SEVERE”).

By default, IIS uses IIS Logging File Format; however, for the system administrator can easily configure IIS to use W3C, NCSA, or custom logging formats instead as per VA conventions. For more information on IIS logging, see the online manual ([www.iis.net/learn/manage/provisioning-and-managing-iis/configure-logging-in-iis](http://www.iis.net/learn/manage/provisioning-and-managing-iis/configure-logging-in-iis) ).

The IRDS web application is currently configured to do its logging through IIS. The system administrator can configure application-specific behavior in the IIS log configuration utility.

#### Network

IRDS can suffer from errors due to network conditions between the client and the server, or between the server and CDW. If there are network problems during the initial loading of a page, the client may display built in error messages (e.g., HTTP 404). On the other hand, if there are errors transmitting data in the background, the client JavaScript will attempt to retry the operation before failing with a user-friendly error message.

For network issues between the server and CDW, if the application can catch and retry the operation, it will. For network errors beyond the application’s grasp, the server will fail and log the operation and redirect the user to a user-friendly error page. The error page typically instructs the user to see the system administrator.

#### Authentication & Authorization

All authentication and authorization errors are caught by the application and logged.

For IRDS authentication errors, the system will prompt the user a total of three times and then lock the user account and provide them with a message to see the system administrator. IRDS authorization errors should be rare, but if they occur, the user will be notified contact their system administrator. The administrator can adjust the user’s settings as needed.

## Dependent System(s)

IRDS is dependent upon CDW for retrieving Veteran PII/PHI data. IRDS is also dependent on VLER Direct to submit secure messages to outreach personnel regarding high risk veterans. For persistent failures connecting to CDW or VLER Direct, there’s nothing that can be done resolve access or connectivity issues within IRDS; all other errors must be resolved in cooperating with CDW or VLER Direct support.

## Troubleshooting

Troubleshooting IRDS issues consists of checking the logs and tweaking configuration settings. Most application behavior cannot be adjusted without modifying code. The following table summarizes the types of errors and resolution procedures likely to occur in IRDS:

| Type | Procedure |
| --- | --- |
| Errors | Check logs and report issue. |
| Database connectivity issues | Check status page, application and database logs and connection string. Troubleshoot using SQL Server Management Studio client. |
| Other database issues | Check logs and report issue. |
| CDW connectivity issues | Check logs, report issue to appropriate help desk. |

Table 23: Troubleshooting IRDS

The first step in most cases is to check the system status page. The process of authenticating and viewing the status page will give you some information about the system stability, because this process exercises the application, database, and CDW. The inability to authenticate or errors reported on the status page allows the system administrator to narrow his/her focus.

The next step is to check the logs. The logging level can be temporarily dialed up in each logging configuration file (see 3.2.1) to support DEBUG-level messaging as needed. The logs will display detailed information about the type of problem that is occurring, and can be tailed and searched. If the application is operational in general, failing actions can be tested in the application and checked in the logs.

If the application cannot connect to the database, the system administrator can check whether the *MSSQLSERVER* process is running, check the SQL Server logs, and test connecting to the database using various parameters using the SQL Server Management Studio client (application interface). The SQL Server Management Studio client can also be used to query and modify data or state as needed in order to resolve the issue.

If the problem is with CDW, the connection information can be changed or confirmed with CDW support technicians. Most CDW issues will require cooperation with CDW support.

If the problem is with VLER Direct, the connection information can be changed or confirmed with VLER Direct support technicians. Most VLER Direct issues will require cooperation with VLER Direct support.

Finally, if the issue lies within the application itself, such as a bug or the inability to deal with an unforeseen issue in the production environment, the application source code can be modified as needed to resolve the issue.

The following tables provide a detailed listing of error conditions and resolution actions.

|  |  |  |
| --- | --- | --- |
| **Category** | **Description** | **Actions** |
| Verify IRDS Account in the Login Page | User is presented with “Account is locked. Please contact System Admin” | User’s account is locked in IRDS. User has entered the wrong username and password more than 3 times. User should seek assistance with VA Active Directory admin to ensure their username and password is correct or have it be reset to something else. Then the user needs to contact the IRDS System Administrator to resolve the locked account issue. |
| Verify IRDS Account in the Login Page | User is presented with: “Max number of sessions reached, Please log out from your active sessions or wait for 30 secs and try again” | Users login attempt is blocked due to max user sessions already opened. User must close active sessions by logging out successfully and try again or wait 30 secs if browser window was closed prior to successful logout. If problem still persists please contact the IRDS System Administrator to resolve the issue. |
| Verify IRDS Account in the Login Page | User is presented with: “Account is disabled due to inactivity, Contact system admin.” | User has entered valid account credentials however it has been more than 30 days since last login attempt. After 30 days of no login activity, user account will be deactivated and will need to contact IRDS System Administrator to resolve the issue. |

**Table 24: Errors and descriptions**

|  |  |
| --- | --- |
| **Category** | **Actions** |
| Unable to Connect | The server cannot be connected to from the device the veteran is using. The network used by the veteran should be checked to make sure it has a network connection (e.g. navigate to a page within the VA’s network). If the device can navigate to another VA site then the server should be checked to make sure it is still running. If it is then IT should be contacted to ensure that the network hasn’t become fragmented. |
| Database Connection Issues | As a system administrator, log into the web server and ensure the web server can communicate with the database, ensure the database is running by checking the Windows service list, and ensure the database account the web site is using is able to be used to log into SQL Server |
| CDW Connection Issue | As a system administrator, log into the web server and ensure the web server can reach the CDW server. Contact the CDW administrator to verify existing connection settings is still valid. |
| VLER Direct Connection Issue | As a system administrator, log into the web server and ensure the web server can reach the VLER Direct server. Contact the VLER Direct administrator to verify existing connection settings is still valid. |

**Table 25: Actions for common errors**

## 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 system crashes or is brought down, it can be simply restarted by restarting the database and IIS server processes and then viewing the application status page. The two processes, which are covered earlier in this document, are run as Windows services and can be started from the Services snap-in. Once the services start, the system administrator can logon to the application and view the status page in order to verify connectivity. The full steps are as follows:

1. Ensure the SQL Server and IIS services are running.
2. Load the application home page
3. View the status page to ensure the application can connect to CDW

If the Windows services do not start properly, the Event Viewer and the log files for each service can be checked for errors. Failures in the services are unlikely to happen, however, if the operating system itself is healthy.

### Restart after Database Restore

The system can be restarted after restoring from a database backup by simply accessing the application. If the application server was taking offline in order to prevent access to the database during the restore, the application server should be restarted as well before utilizing the application.

### Back Out Procedures

The upgrade back out procedure consists of notifying the service desk, taking the application offline such that users see a “down for maintenance” page, performing the back out steps (restoring the database, redeploying the old version of the application, etc.), checking the application locally, restoring service, and then notifying the help desk that the maintenance is over.

Notifying the service desk should be done in accordance with the policies established at the time. (Currently, there is no service desk for supporting IRDS.) This may take the form of an email, call, or service desk ticket.

Taking the application offline consists of changing configuration so that all requests except those passing in a special maintenance parameter (established by the system administrator) are routed to a simple HTML page that contains the designated “down for maintenance” message instead of hitting the web application.

After all existing requests to the web application have ended the old version of the database or application can be restored as per the back-out plan. This configuration can be verified by using another maintenance parameter to access the application and view the status page.

Once the back-out has completed, the routing expression on the server is restored to point to the desired version of the web application. The full steps are given below:

1. Notify the Service Desk about back-out plan initiation via email or service ticket
2. Disable user access to the system by engaging the maintenance mode routing expression
3. Restore backup taken before the change implementation by following the database restore procedures in this document
4. Conduct system health checks by utilizing the maintenance mode health check parameter and viewing the status page
5. Enable user access by disengaging the maintenance mode routing expression
6. Notify the Service Desk of successful back out

For more information on how to configure the maintenance routing expressions, see the IIS manual: <http://technet.microsoft.com/en-us/library/cc770409(v=ws.10).aspx> .

# Operations & Maintenance System Support

An understanding of how IRDS is supported by various organizations within the VA is important to operators and administrators of the system. If you are unable to resolve an issue, then it is necessary to understand how to obtain support through OI&T’s system support organizations. The following sections describe the support structure and provide procedures on how to obtain support.

## Support Structure

This section describes the systems support structure as seen from the perspective of operations personnel. The first section defines the support hierarchy through which a support request may navigate. The second section defines the responsibilities for each level of support.

### Support Hierarchy

There will be two levels of production support for IRDS until the application achieves nationwide deployment. The first level will consist of triage, account management, and basic troubleshooting performed by a Healthcare System Technical Administrator (system administrator). The second level will consist of application code and database change management as described within the IRDS Change Management Guide.

Following nationwide deployment, it is expected that the application will migrate to standardized VA support and change management practices, with tier 1 support performed by the National Service Desk, tier 2 support performed by VA regional IT support staff, and tier 3 performed by application developers as designated by IRDS program management.

### Division of Responsibilities

This section defines the scope and responsibilities of each support tier.

## Support Procedures

The IRDS support procedures will consist of triage, troubleshooting, and change management.

1. Defect and change requests triaged by Program Administrator
2. Troubleshooting by Healthcare System Technical Administrator
3. Change management performed by application developers as authorized by Change Control Board

1. Triage: The Program Administrator will collect and triage application defect and change requests from users. These requests will be entered in the IRDS change management backlog in the form of trouble tickets.

2. Troubleshooting: The program administrator will assign trouble tickets to the Healthcare System Technical Administrator (HSTA), who will analyze, troubleshoot, and document the reported issues. If the HSTA can resolve the issue through at the configuration or database level, or through coordination with the National Service Desk (in the event of a CPRS or VistA issue), the HSTA will document the resolution within the ticket and mark it resolved.

3. Change management: If HSTA is unable to resolve an issue without modification of the application source code, the HSTA will change the ticket state to needing Change Control Board (CCB) review. The CCB, which will consist of the VA PM, Program Administrators, Healthcare System Technical Administrators, and designated VA IT/support staff, will prioritize and assign all application change requests to designated application developers. The application developers will estimate the amount of time needed to complete the work associated with the ticket, and the PM will allocate the ticket to a specific development sprint. Once the application development team completes and tests the work, they will mark the ticket resolved and perform the application release as authorized by the CCB.

For a detailed explanation of change and defect management, see the IRDS Change Management Guide.