

User's Guide

*For the Joint Network Defence and Management System
(JNDMS)*

Prepared By:
Scott MacDonald
MDA Systems Ltd
Suite 60, 1000 Windmill Road
Dartmouth NS, B3B 1L7

MDA Reference # DN1009, Issue 1/1
Contract Project Manager: Brett Trask, 902-481-3511
PWGSC Contract Number: W7714-040875/001/SV
CSA: Marc Gregoire, Project Manager, 613-998-2113

The scientific or technical validity of this Contract Report is entirely the responsibility of the Contractor and the contents do not necessarily have the approval or endorsement of the Department of National Defence of Canada.

Contract Report # DRDC-RDDC-2014-C81
October 2013

Principal Author

Original signed by Scott MacDonald

Scott MacDonald

Project Engineer, JNDMS

Abstract

This document provides a guide to the usage and administration of the Joint Network Defence and Management System (JNDMS).

Résumé

Le présent document constitue un guide de l'utilisation et l'administration du Système interarmées de défense et de gestion des réseaux (SIDGR).

Executive summary

User's Guide: For the Joint Network Defence and Management System (JNDMS)

Introduction or background: The Joint Network Defence and Management System (JNDMS) is a Technology Demonstrator to evaluate the Situational Awareness of networks. The JNDMS provides a Web Portal for operators to use and explore the network and events or circumstances that impact their network.

Sommaire

User's Guide: For the Joint Network Defence and Management System (JNDMS)

Introduction ou contexte: Le Système interarmées de défense et de gestion des réseaux (SIDGR) constitue un démonstrateur de technologie visant à évaluer la connaissance de la situation en réseau. Le SIDGR fournit aux opérateurs un portail Web qui leur permet d'utiliser et d'explorer le réseau et les événements et situations qui ont des répercussions sur le réseau.

Table of contents

Abstract	i
Résumé	ii
Executive summary	iii
Sommaire	iv
Table of contents	v
List of figures	vii
List of tables	viii
1 Introduction.....	1
1.1 Purpose	1
1.2 Scope	1
2 System Overview	2
3 Server and System Components	3
3.1 NIO Setup.....	3
3.2 Development Lab Setup	4
3.3 Laptop Setup.....	4
3.4 Execution of Core Components.....	5
3.5 Execution of Integration Components	5
3.5.1 JSS Client.....	6
3.5.2 Subnet Creator Tool	11
4 Portal.....	14
4.1 Overview	14
4.2 Navigation	17
4.2.1 Focus and Filters	19
4.2.2 Navigation Tree Options	20
4.3 2D Map.....	22
4.3.1 Map Icons.....	26
4.4 3D Map.....	27
4.5 Data View	30
4.5.1 List Views	30
4.5.2 Detailed Views	36
4.6 Graph View	45
4.7 Search and Filters	48
5 Administration and Trouble Shooting	50
5.1 System Start Up and Shut Down	50
5.1.1 Oracle Database	50
5.1.2 Web Applications.....	50

5.2	System Issue Diagnosis	51
5.2.1	Portal	51
5.2.2	Core Services	51
5.2.3	Database	52
5.2.4	Web Application Server	52
5.2.5	Common Issues	53
5.3	Database Maintenance	57
5.3.1	Enterprise Manager	57
5.3.2	JNDMS Datasets	58
	List of symbols/abbreviations/acronyms/initialisms	60

List of figures

Figure 1: System Login	17
Figure 2: Portal Overview	18
Figure 3: 2D Map	22
Figure 4: 2D Map operation popup.	23
Figure 5: 2D Map Asset popup	23
Figure 6: 2D Map Icon Aggregation.	24
Figure 7: 3D Map	27
Figure 8: 3D Map Options.....	28
Figure 9: 3D Map Operation popup	29
Figure 10: 3D Map Icon Cluster.....	29
Figure 11: List View Column Filters.....	30
Figure 12: List View Column Selection	31
Figure 13: List View Paging Toolbar	31
Figure 14: List View Toolbar with Edits.....	31
Figure 15: Tree List Views	32
Figure 16: Detail Views.....	36
Figure 17: Graph View Applet.....	45
Figure 18: Graph View Loading Initial Roots.....	46
Figure 19: Graph View Entity Expansion	46
Figure 20: Graph View with Dependent and Redundant Links.	47
Figure 21: Text Search	48
Figure 22: Date Search	49
Figure 23: Search by IP Address	49

List of tables

Table 1: Development Lab Configuration..... 4

Table 2: JSS Client Command Line options 6

Table 3: Map Icons..... 26

Table 4: Common Issues and Solution..... 53

1 Introduction

1.1 Purpose

This document will provide a basic guide for the use and trouble shooting of the Joint Network Defence and Management System (JNDMS) under contract W7714-04-0875/001/SV.

1.2 Scope

The following sections comprise this document:

- System overview (section 2). This section will identify the core components of a running JNDMS.
- Server Components (section 3). This section identifies server or system component and their usage.
- Portal (section 4). This section outlines the use of the portal through a web browser.
- Administration and Troubleshooting (section 5). This section provides some information to help trouble shoot issues and administer the system.

2 System Overview

The JNDMS provides a prototype ability to monitor a network and evaluate Situational Awareness. The system provides this ability by leveraging enterprise management and security management tools and providing additional tools for analysis and to combine all available information so that a user can explore the relationships.

The JNDMS is made up of several core components or services. These are:

- Enterprise Information Management. This component leverages existing enterprise tools to provide the initial discovery of the network topology as well as periodic updates and monitoring of the network.
- Security Management. This component leverages existing enterprise security tools to provide integration to a network's security infrastructure, to collect security events, to provide an initial analysis of these events and to escalate to the core of JNDMS where required.
- Integration tools. Many different tools and sensors on a network could provide essential information to JNDMS. There are a number of tools that provide integration points to the required sensors.
- Data warehouse and data transformation services. Much of the ability of JNDMS to provide a single view of the network and to combine the information from disparate sources is because there is a central warehouse in which all information and relationships are stored. These components are comprised of the database itself as well as various tools used to translate available information into a formation useful to JNDMS.
- Core system services including data sharing. At the core of JNDMS there is the core services (JNDMS System Services – JSS) that provides system I/O, pre-processing and interacts with the analysis component and data sharing.
- Analysis (decision support). This component is responsible for analysis of the information that has been gathered to provide some key indicators for the Situational Awareness.
- Presentation. This component provides the end user's experience through a web browser.

Each of the above components or services can be configured in many different ways to provide support for various network configurations. The following sections will identify the components used during the DREnet deployment efforts as part of the JNDMS Technology Demonstration project.

3 Server and System Components

A running JNDMS consists of a number of core components that are mandatory for any deployment. A JNDMS may also be comprised of one or more optional components that vary depending on the tools or sensors available in any given network configuration.

The core services that are mandatory consist of the following:

- The data warehouse.
- The JSS/DSS.
- The portal.

The above are essential to the running of any JNDMS, however they do not provide any interaction or monitoring of the network. In addition to the above the following are components that would typically be found:

- Network infrastructure monitoring. This was provided using CA Spectrum.
- Security event monitoring. This was provided using Intellitactics Security Manager.
- Software inventory management.

The following sections identify how the above services have been deployed or setup in three different environments.

3.1 NIO Setup

The DREnet deployment effort during the Technology Demonstration project was setup in the NIO lab. This setup was to be able to monitor the live DREnet and to provide a prototype capability for this network. Because of the nature of the live data this setup had to accommodate concerns such as stability and privacy. In no way could JNDMS be capable of interfering with the normal operation of the network. This setup also required split responsibilities. Several of the sensors or collection tools would be managed by DRDC directly without support from the project team.

In this configuration there were two separate labs setup for JNDMS, as well as access to databases at remote sites. The first lab would be accessed and run by the project team with DRDC while the second lab would only be access by DREnet management.

The first lab consisted of the following:

- Web application server.
- Database server.
- Intellitactics server.
- ♦ Development server.

- ♦ Demonstration workstation.

The second lab consisted of:

- Spectrum installation.
- Integration box.

3.2 Development Lab Setup

The development lab was setup to support building and testing of JNDMS. This environment, shown below, provided the core servers, support for historical components (for example ArcGIS) as well as sample installations of enterprise tools.

Table 1: Development Lab Configuration.

Host	Location	IP	OS	Primary Roles
HALIFAX				
Overseer	Halifax	142.128.80.140	Redhat ES 4	DNS
jndms_svn	vm - Overseer	142.128.80.142	Fedora Core 4	SVN, Xwiki,Bugzilla
Gatmaster	Halifax	142.128.80.150	Redhat ES 4	Oracle DB, ISM
caump01	vm - Gatmaster	142.128.80.192	Windows 2003	Admin
Protector	Halifax	142.128.80.160	Windows 2003	ArcGIS,Portal
WatchDog	Halifax	142.128.80.180	Fedora Core 4	VMWare Server
cauni01	vm - WatchDog	142.128.80.189	Windows 2003	Unicenter
Shield	Halifax	142.128.80.190	Windows 2003	Hudson-Artifactory-Routers
Spectrum	vm - Shield	142.128.80.191	Windows 2003	Spectrum
eHealth	vm - Shield	142.128.80.193	Windows 2003	eHealth
Breakdown	Halifax	142.128.80.200	Redhat ES 5	Backup/replacement JNDMSSVN

3.3 Laptop Setup

A typical laptop setup will have the core components of JNDMS installed in one place. This would include the following:

- Web application server. This would be Tomcat running the JSS/DSS as well as the portal.
- The database server.

3.4 Execution of Core Components

This section identifies how each of the core components is executed in a typical JNDMS environment.

- Database. The database in JNDMS is Oracle and is generally set to start on boot up. This should be always available.
- Web application server. Tomcat is used as the application server and, by default, resides in c:\jndms\liferay. The Tomcat installation includes the following:
 - ♦ The Liferay portal. This is used to run the older JNDMS portal.
 - ♦ The JUI Web Application. This provides both the server and client side components of the older JNDMS portal.
 - ♦ The JNDMS Portal Web Application. This provides both the server and client side components of the current JNDMS Portal (as described in this document).
 - ♦ The JSS Web Application. This provides the server side components of the JSS and the DSS.

Tomcat is generally run by its own startup and shutdown scripts (batch files) found in c:\jndms\liferay\bin.

3.5 Execution of Integration Components

This section identifies how the supplementary or integration components are executed. Most of the integration tasks, including monitoring of IP360 and Centennial databases, are through the use of the JSS Client (see section below).

The other sub-section identifies how to run the subnet creator. This tool is used to help build a simulated environment and is not required for normal operation.

3.5.1 JSS Client

The JSS Client is a java application that provides much of the initial contact with external systems and sensors. This client provides a tool that can communicate with the JSS (core of JNDMS) to provide updates or new information.

This tool is run from `c:\jndms\bin\jss_client.jar`. The command line options can be found in the table below:

Table 2: JSS Client Command Line options

Operation	Parameters	Notes
SIM	Type	Classifies type of event
	Sub type	More detailed classification of event, if available
	Source ip	IP address of the source of the event
	Target ip	IP address of the target of the event
	Sensor ip	IP address of the sensor recording the event
	Sensor event ID	An identifier that can be used by the referenced sensor to identify this event. This identifier is likely meaningless outside of the context of the sensor.
	Sensor CVE ID	One or more vulnerability references that are relevant to this event.
	Base Priority	The SIM's recommendation for the priority for this type of event.
	SIM Priority	The SIM's recommendation for the priority of this event after correlation.
	Correlation CVE	Additional vulnerability references.
	Sensor Time	Time of this event as recorded by the sensor.
EIM	Type	Type includes ip_up, ip_down, ip_change, new_interface, link_info and alarm.
	Source	Identifier for which component is reporting.

Operation	Parameters	Notes
	Source_inst	Used in combination with the source field to uniquely identify the source product and particular reporting agent or instance.
	Zone_id	This is used to identify which network zone the agent is reporting from. This is used to ensure there are no ambiguities in cases where network address translation is being used.
	IP Address	IP address for event. This will refer to the source or current IP addresses for event types that require multiple addresses, such as source and destination.
	Name	This is the name of the entity being reported on and depends on the type of event. It will refer to the source or current address and generally will be the host name.
	Label	This is a label applied to the entity being reported on. It refers to the same entity as the IP address and name fields.
	UUID	This is an identifier for this entity (same as IP address and name fields) that is unique for this source.
	Class	The field depends on the source that is reporting and may be used to remove ambiguity in reports.
	Create Date	Date and time of event.
	Alt IP	This is an alternate IP address used by some event types. It will refer to the destination or previous address related to this event.
	Alt Class	This is the class of the entity (source dependant) referenced by the alt IP, alt name and alt uuid fields.
	Alt UUID	This is the unique identifier (source dependant) for the entity referenced by the alt IP, alt name and alt uuid fields (some or all may be present).

Operation	Parameters	Notes
	Alt Name	This is the name for the alternate entity.
	Alt Label	This is the label given to the alternate entity.
	Severity	This allows the reporting agent to identify its assessment of the severity of the event.
	Status	This allows the reporting agent to identify its assessment of the status of the events, especially when reporting alarms.
CME	Source	An identifier for the source of this information.
	URL	The URL to the XML data. The format must meet the schema for the CME (http://cme.mitre.org/).
CVE	Source	An identifier for the source of this information.
	URL	The URL to the XML data. The format must meet the schema for the CVE (http://nvd.nist.gov/).
Safeguards	Source	An identifier for the source of this information.
	URL	The URL to the safeguard data
Operations	Source	An identifier for the source of this information.
	URL	The URL to the operations data.
Assets	Source	An identifier for the source of this information.
	URL	The URL to the asset data. This data must be formatted according to the source.
Dp_update	Source	An identifier for the source of this information.
Dp_enable	Source	An identifier for the source of this information.
Dp_disable	Source	An identifier for the source of this information.
V_scan	Source	An identifier for the source of this information.

Operation	Parameters	Notes
	URL	The URL to the XML formatted vulnerability scan.
Cap (Common Alerting Protocol)	Source	Identify the source of the event.
	URL	The URL to the XML event as defined by the Oasis (http://www.oasis-open.org/home) schema.
Jndms_xml	Source	Identify the source of the information.
	URL	The URL to the XML event as defined by the JNDMS reporting schema. This event type can be used to report on many JNDMS relationships including assets, operations, services and the dependencies between them.
Scan_dir	Source	Identify the source of the information.
	Dir	Directory to scan for incoming events (used in the sharing scenarios)
	interval	The time interface to wait before checking the directory for new events.
Scan_ip360		This option will scan an IP360 database for changes to be submitted to JNDMS.
	Source	Identify the source of the information.
	Dir	Optional directory to write events to.
	noJSS	This is a flag to indicate that the client should not send the resulting information to JNDMS (through the JSS). This is used for testing and validation purposes.
	lastImport	This identifies the last import that was done. This can be never to get all vulnerability records.
	minDelay	A delay can be set between scans. This can be used to ensure the database is not scanned too often.

Operation	Parameters	Notes
Scan_discovery		This option will scan a Centennial discovery database for changes to be submitted to JNDMS.
	Source	Identify the source of the information.
loadSpectrumTopology		This will load an XML status report from Spectrum and send it to JNDMS. This is used to provide the entire topology as one event.
	Source	
	url	
Scan_discovery		This will scan a given directory for Spectrum XML reports. Any reports found will be sent to JNDMS.
	Source	
	dir	
	Interval	
Bulk_discovery		This will scan a Centennial Database and send all software inventory records to JNDMS.
	Source	
	sourceAddress	
View	Ids	A list of event IDs to view.
	Type	‘summary’ or ‘all’ identifies what information is provided for each event.
Remove	Ids	This will remove the given ID from the queue.
[all]	saveEvent	Any event can be saved for later play back. This parameter is set to true (saveEvent=true) or false.
	Dir	If events are stored, this is the directory they will be stored in.

In addition to the command line parameters a properties file is also used (contained in the jar file):

- Database credentials for IP360
 - ♦ jndms.ip360.db_driver
 - ♦ jndms.ip360.db_url

- ♦ jndms.ip360.db_user
 - ♦ jndms.ip360.db_pass
- Database credentials for Centennial
 - ♦ jndms.centennial.db_driver
 - ♦ jndms.centennial.db_url
 - ♦ jndms.centennial.db_user
 - ♦ jndms.centennial.db_pass
 - ♦ jndms.centennial.db_version
- Security certificate configuration
 - ♦ security.client.keystore.path
 - ♦ security.client.keyStoreType
 - ♦ security.client.keystorePassword
 - ♦ security.client.truststore.path
 - ♦ security.client.truststorePassword
- Location of JSS (if not provided on command line)
- jndms.deploy.jss_url

3.5.2 Subnet Creator Tool

The Subnet creator is a tool written in Java to support the automated creation of simulated subnets. It can be run giving it a subnet, the number of hosts to create, the link from the new subnet to the network and a profile of software that should be installed on new hosts. This tool was used to scale the simulated network and can also be used to create subnets with specific profiles.

Subnet Creator command line:

```
> java -jar SubnetCreator.jar -j [path to client] -e [JSS] -s [template] -oh [hardware] -os
[software] -nh [number] -r [subnet]
```

Description of options:

1. Path to client: This is the path to the JSS client JAR (jss_client.jar).
2. JSS: This is the URL of the JSS endpoint that will be used to submit the generated assets.

3. **Template:** This is a file that describes what software assets will be created on the newly created hosts. This is an XML file that conforms to the JNDMS XML schema (see the JNDMS Design Document, rev 3.1, section 3.6.2.3.1) to define assets.

The following is an example of a software template that will create 'win2k' (Windows 2000) on all newly created workstations.

```
<?xml version="1.0"?>
<jndms xmlns="http://jndms"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://jndms jndms.xsd">
  <assets>
    <!--create placeholder reference -->
    <existingAssets>
      <assetRef>
        <id>workstations</id>
        <ip_range_data/>
      </assetRef>
    </existingAssets>
    <!--list of assets and products to create on workstations -->
    <asset create_on_all="true">
      <method>software</method>
      <name>win2k</name>
      <host>
        <id>workstations</id>
      </host>
      <product create="true">
        <name>Windows 2000</name>
        <vendor>Microsoft</vendor>
        <version>Professional SP2</version>
      </product>
    </asset>
  </assets>
</jndms>
```

4. **Hardware:** This is the output file that will store the list of newly created hardware assets. The file that is created can be submitted directly to JNDMS using the following command:

```
> java -jar jss_client.jar com.mdacorporation.jndms.JSS.Client.JSSBatchClient [file]
```

5. **Software:** This is the output file that will store the list of newly create software assets. This file can be submitted directly to JNDMS using the following command:

```
> java -jar jss_client.jar op=jndms_xml source=subnetcreator url=file:[file] endpoint=[jss url]
```

6. **Number:** This is the number of hosts to create. If this is not given the entire subnet (default of 90.1.1.0/24) will be created.
7. **Subnet:** This is the subnet mask to use. For example 192.168.3.0/24.

A convenience script is provided in Test/cycle_3/Discovery_scripts called make_subnet.bat that combines the above steps of creation and submission. It can be called in the following manner:

1. Create a given number of assets starting at 90.1.1.0:

```
make_subnet.bat -nh <num_assets>
```

2. Create a full subnet range:

```
make_subnet.bat -r <subnet_range(e.g. 90.1.1.0/24)>
```

3. Create a given number of assets starting at a subnet range:

```
make_subnet.bat -nh <num_assets> -r <subnet_range(e.g. 90.1.1.0/24)>
```

4 Portal

4.1 Overview

The JNDMS portal is a web interface for users of JNDMS. This portal provides the usage and interaction with the system other than actions required by a system administrator.

There were two separate portals developed during the final phases of the Technology Demonstrator, each of which is packaged as a Web application ARchive (WAR). This section documents the version developed for the DREnet deployment efforts built using the Google Web Toolkit (GWT).

The section below on Navigation (section 4.2) identifies how to navigate within the portal from the initial login. The sections beyond identify the information available within the different views of JNDMS, such as the 2d map the 3d map, the data view and the graph view.

To understand how the navigation works and to understand much of the content within the portal you must understand how JNDMS presents its core concepts or entities. The following is a brief outline of the JNDMS entities:

- **Operations.** This identifies an organizational unit with some requirements on the IT infrastructure. This was initially modelled after operations with DND, however was quickly extended to include any identified organization unit. This would, of course, include DND operations, however it would also include support services and other groups that have to be separately identified.
- **Assets.** Assets within JNDMS identify any physical or logical item that is tracked or has an identified dependency. This includes network infrastructure (switches, routers, etc), computer equipment (servers, workstations, peripherals, etc), software, and services (email, portals, etc).

Each asset has a type and category associated with it. The type identifies the following:

- ♦ **Primary hardware.** This represents hardware assets that generally stand on their own, with their own dependencies. This would represent the actual host for example in a computer configuration. The peripherals and software should all be associated with the primary hardware. Examples of primary hardware include hosts, servers, workstations, routers, switches, firewalls, etc.
- ♦ **Secondary hardware.** This represents hardware that is a peripheral or component of a primary hardware item. An example of this found in all JNDMS installations is the network card. This is tracked separately to allow assets to have multiple IP addresses, however each one must be associated with a primary hardware (its host).
- ♦ **System service.** This identifies software or services that are generally local to a given computer. This would include most locally installed software and the operating system.

- ♦ **Network service.** This identifies software or services that are shared across the network. This type is generally used when creating a logical server.

The category will expand upon the type and identify if this asset represents software a switch, a host or other component.

Each asset can also have an associated product. These values can track the vendor, the product and the version.

- **Vulnerabilities.** Vulnerabilities within JNDMS generally represent software vulnerabilities, such as those published by the National Vulnerability Database (NVD), but may also include other issues such as physical vulnerabilities.

Vulnerabilities that are tracked by security firms and are published against a list of vulnerable products are known as vulnerability definitions in JNDMS. These list the details of the vulnerability itself.

In addition to the vulnerability definition, in JNDMS, you may also have one or more vulnerability instances. A vulnerability instance is a specific asset that is vulnerable to an identified vulnerability.

- **Events/Incidents.** Events within JNDMS consist of any activity that could have a potential impact. When events, such as reporting new vulnerabilities or intrusion attempts, are analyzed their impact is shown within JNDMS. If there is an impact then the event would be classified as an incident.

Events and incidents should, throughout their normal life-cycle, be eventually resolved or mitigated. Only active events will be considered as part of the impact analysis.

- **Safeguards.** Safeguards, in JNDMS, are any method or item that could protect you from one or more vulnerabilities. These are generally software safeguards, however they could be extended into the physical domain.

Safeguards in JNDMS are of two general types. They can either be perimeter safeguards or they could be very focused. Perimeter safeguards could be devices such as firewalls or Intrusion Detection Systems that don't protect individual hosts, they protect or monitor traffic between network zones. Focused safeguards on the other hand, generally are targeted at specific vulnerabilities. An example of a focused safeguard would be a virus scanner or security patch.

Safeguards, like vulnerabilities, are generally divided into the safeguard definition (generally referred to as just the safeguard) and specific instances of safeguards. In this case a particular patch would be the safeguard and it could have been applied on a number of assets.

- **Locations.** Locations within JNDMS can be quite generic, such as a point on the map, or they can be very specific and include addresses, even room numbers. Each location is meant to be quite flexible and allow the end user to choose the level of granularity that they want to track.

- **Networks / Zones.** A network or zone within JNDMS is a collection of subnets. Many devices such as firewalls will provide policy enforcement points between zones and any filtering device will require a new zone to be created. Zones within JNDMS also allow more human readable names to be associated with a collection of common subnets.
- **Points of Contact (POC).** This entity tracks potential points of contact that may be related to one or more other entities. A network, for example, may have several points of contact related to maintenance, security or policy. This allows for each of these to be collected and explored as issues arise.
- **RFCs.** A Request For Change (RFC) is a formal request to have a specific action taken on a network. These are often used to implement patches or upgrades. This allows the tracking of active RFCs as they relate to the JNDMS view of the network.

As well as the core entities there are also several indicators that are used to help with evaluating Situational Awareness and to present the analysis done by JNDMS. These are:

- **Risk.** The risk score is the primary ranking factor identified for quick assessment of Situational Awareness. This value is based on the relationships between the assets and the operations. Every event analyzed by the system will examine the impact each event has on every asset, then identify the relationship that each asset has with each operation.

In JNDMS risk must be associated with an operation. If there is no link between an asset and an operation then that asset cannot contribute to the overall risk. This does not, however, imply that every asset must explicitly be identified by every operation. At the core of JNDMS is the idea of relationships between assets as well as relationships between assets and operations. The combined impact is evaluated as part of the risk.

An example of the above would be an operation that depends on email. The infrastructure would identify that email is provided by one or more servers and that there are one or more clients required to access the service. The operation would then identify that this service is important, critical or less important. This core information, along with up to date information on network topology and vulnerabilities would allow us to analyze any new event that might impact any of the above servers or workstations. This analysis would also include the examination of other potentially affected computers on the same network or possible communication outages between the client and the servers. Any of the above could escalate the risk.

The risk is primarily based on the relationships provided as well as the events and vulnerabilities that exist on the network. The availability of an asset or service can also be included in the risk value.

The risk scores are summarized as low (green), medium (yellow) and high (red) throughout the portal.

- Incidents. The number of incidents can be used to assess activity on the network. An initial analysis of the incoming events is done to determine if there is any identified impact as a result of this event. Any event with impact is considered an incident.
- Availability. The availability of assets can also be an indicator of potential issues.

4.2 Navigation

The navigation within the JNDMS portal starts with the login page (see Figure 1). All access to the portal and associated web services depends on having an appropriate username and password. For added security the portal can be deployed and accessed over Secure Sockets Layer (SSL).

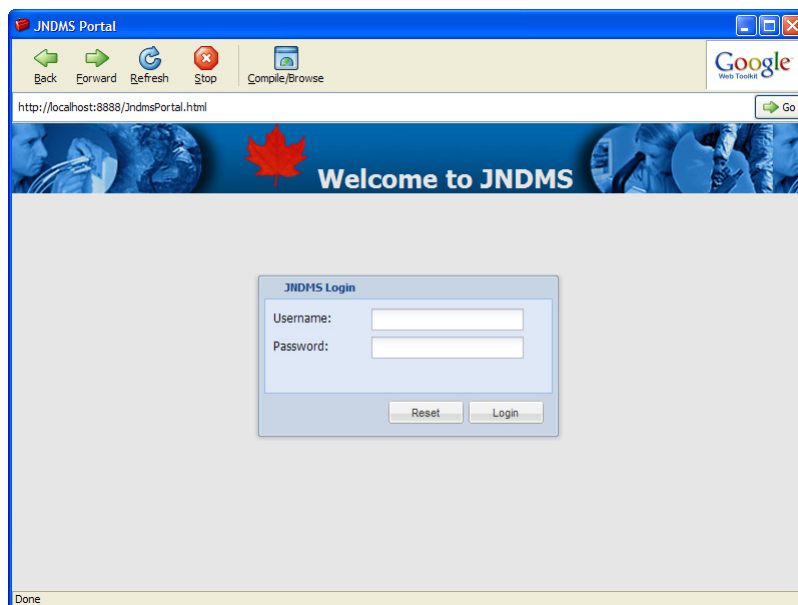


Figure 1: System Login

After the user logs in they are presented with the JNDMS portal. This portal is a web page and consists of four main areas of interest (see Figure 2).

Figure 2: Portal Overview

The following is a description of the areas used by the portal interface (see Figure 2):

1. **Side bar.** The side bar consists of an accordion style widget. The sections available in the current portal include the following:
 - ♦ **Navigation.** This provides a tree view in which each folder or leaf is a link that will alter the content of the other views. See below (4.2.2) for details.
 - ♦ **Search.** This provides access to the search dialog. See section 4.7 for details.
 - ♦ **Filter.** This provides access to the filter dialog. See section 4.7 for details.
 - ♦ **Settings.** This is a place holder in the user interface for values that would be configurable from the user interface.

2. **Primary View.** The primary view provides the largest content viewing by default. It consists of a tab panel so that the presentation of the current view (see section on focus and filters, 4.2.1, for details) could be changed. The four available presentations are:
 - ♦ 2D map. Section 4.3.
 - ♦ 3d map. Section 4.4.
 - ♦ Data view. Section 4.5.
 - ♦ Graph view. Section 4.6.
3. **Secondary data view.** This view contains a data view (see section 4.5) that will respond to events from the navigation tree or the primary view to alter the current view.
4. **Global Status.** This part of the portal contains four graphs that identify key indicators over the past twenty four hours. The indicators are:
 - ♦ Operational risk. Identifies any operations at risk.
 - ♦ Asset availability. Identifies any assets that are off line.
 - ♦ Location risk. This shows any location with elevated risk. A location is considered to have elevated risk if any assets at that location are contributing to operational risk.
 - ♦ Incidents. Shows the history of incidents over the past 24 hours.

4.2.1 Focus and Filters

The core of the navigation within the portal is based on a defined focus and one or more filters. The focus is the base entity that you are interested in viewing (see section 4.1, for details). For example if you click on the 'Operations' folder in the tree navigation your focus will be operation.

There are two types of filters that are applied to each view. The first is the page filter. This is generally created as part of the link that defines the current view. For example if you click on 'Operations / By Location / Ottawa' you will be viewing the operation focus with a location filter applied. Only operations in Ottawa will now be shown.

The second type of filter is the global filter. This is created using the 'filter' panel in the side bar. This type of filter is applied to every view, no matter what the focus or the page filter would be. As an example you may create a global filter based on operations named 'Support Services'. You may then view 'Assests / By Location / Ottawa'. The combined results would only show assets in Ottawa that are related to the Support Services operation.

Another special case of the focus is used to represent detailed pages. Most of the links provide a list view (see section 4.5.1), however many of the links within the data views or popups in the maps provide links to a specific item. In these cases a detailed view (see section 4.5.2) is shown.

4.2.2 Navigation Tree Options

The following options are available through the side bar:

- **Operations.** This will list all operations.
 - ♦ By Name. This will list each operation by name. Each name links to a detailed view.
 - ♦ By Location. This will show a list of operations filtered by the given location.
 - ♦ By Zone. This will show a list of operations filtered by the given zone.
- **Assets.** This will list all assets.
 - ♦ By Location. This will list all assets filtered by location.
 - ♦ By zone. This will list assets filtered by zone.
 - ♦ By operation. This will list assets filtered by operation.
 - ♦ Network View. This will list assets related to the core of the network infrastructure. This is generally used in conjunction with the graph view to get a quick idea of network topology.
 - ♦ Products. This will show a product list.
 - ♦ By Category. This will show assets filtered by category
- **Vulnerabilities.** This will show the list of vulnerability definitions.
 - ♦ Definitions. This will show the list of vulnerability definitions.
 - By Location. This will show the vulnerability definitions filtered by location
 - By Operation. This will show the vulnerability definitions filtered by operation.
 - By Zone. This will show the vulnerability definitions filtered by zone.
 - ♦ Instances. This will show the list of specific vulnerability instances.
 - ♦ Exploits. This will show a list of associated exploits.
- **Events.** This will show the list of all events
 - ♦ By Location. This will show the events filtered by location.
 - ♦ By Zone. This will show the events filtered by zone.
 - ♦ By Operation. This will show the events filtered by operation.
- **Safeguards.** This will list the safeguards.
 - ♦ By Location. This will list the safeguards filtered by location.
 - ♦ By Zone. This will list the safeguards filtered by zone.
 - ♦ By Operation. This will list the safeguards filtered by operation.
- **Locations.** This will list all location.
 - ♦ By Name. This will provide direct links to location details.

- ♦ By Zone. This will show locations filtered by zone.
- ♦ By Operation. This will show locations filtered by operation.
- **Network.** This will show zones.
 - ♦ By Name. This will provide direct links to the zone details.
 - ♦ By Location. This will show the zones filtered by location.
 - ♦ By Operation. This will show the zones filtered by operation.
 - ♦ Zone Borders. This will show zones and zone borders. This is generally used with the graph view to get a quick logical view of how the zones are related, including the borders between them.
- **Points of Contact.** This will show all points of contact.
- **Tools.** There are no links to external tools at this time. This option is to provide direct access to web interfaces related to JNDMS. This would include, for example, links to Intellitactics, Spectrum, the National Vulnerability Database (NVD) as well as others.
- **Reports.** There are no preconfigured reports at this time. This is to provide a link to preconfigured reports.
- **RFCs.** This will show a list of RFCs.

4.3 2D Map

The 2D map provides a basic topographical map (see Figure 3) in which any location or geographical information can be overlaid.

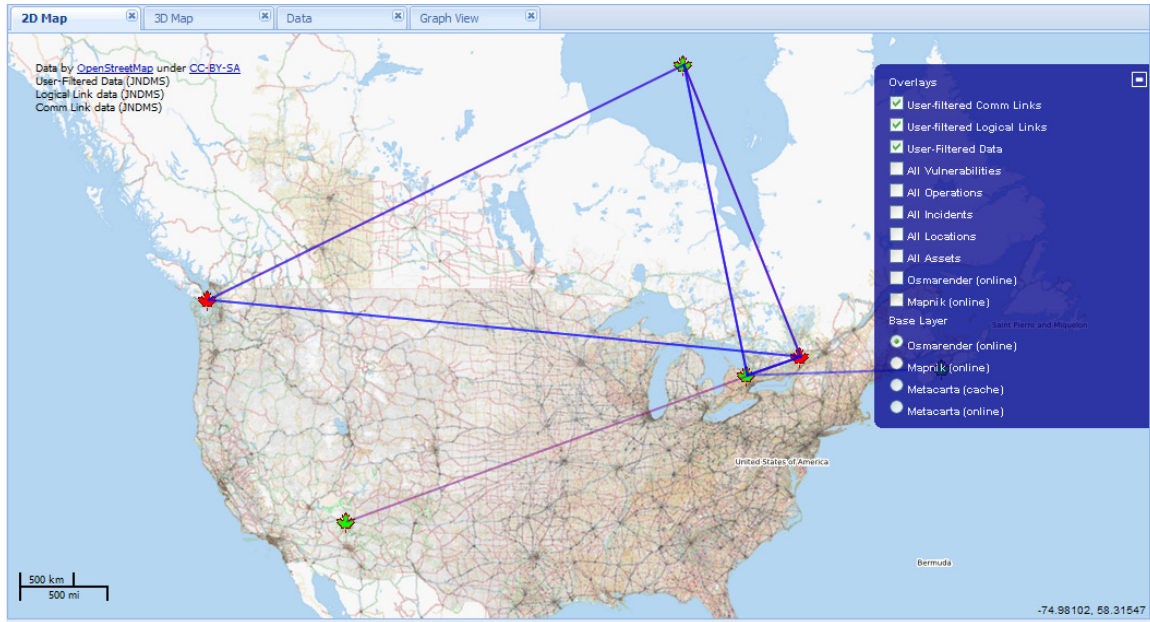


Figure 3: 2D Map

The map consists of a base layer and one or more overlays. The icons and links on the map are clickable and the map component itself provides some navigation. The mouse can be used to:

- Pan: Click and drag
- Zoom: Double click to zoom in. Mouse wheel zooms and when the mouse is used with the shift key a bounding box can be defined.

All information that is displayed on the map is based on the relationships to locations. In some cases the location information of an entity cannot be known and in that case there is an 'unknown' location. This allows all items to be placed on the map and possibly showing their relationships, even if we don't have location information for all items.

When an icon or a link is clicked then a popup will be shown (see Figure 4 and Figure 5). The information in each of the popups will contain information about the item that was selected. Some of the information within the popups will contain hyperlinks. These links can be clicked and will result in detailed information being shown in the secondary view.

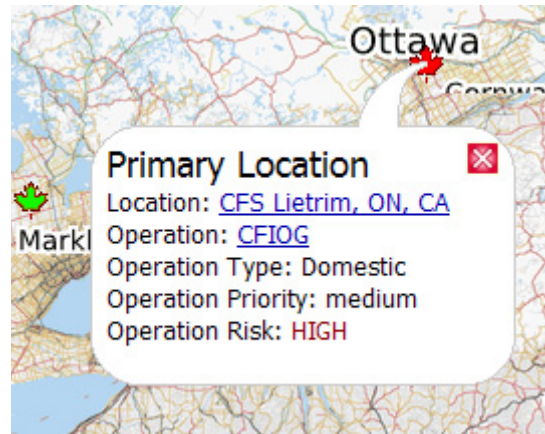


Figure 4: 2D Map operation popup.

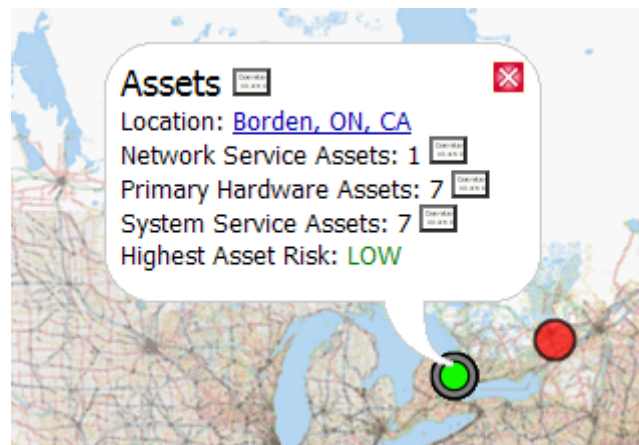


Figure 5: 2D Map Asset popup

In some cases, depending on your current view, there may be many icons in one area. In this case it can be quite difficult to see what icons are there and it also leads to important activity being hidden or overlapped by other items.

The 2D map provides an aggregation feature in which icons that are too close are combined into an aggregate icon (see Figure 6). These aggregate icons have a size loosely based on the number of icons that had been combined. The colour of the aggregate icon is based on the highest risk associated with any of the icons that had been combined.

When an aggregate icon is selected it will show the list of icons that had been combined. To view the details of the associated entities the user must zoom in to separate the icons.

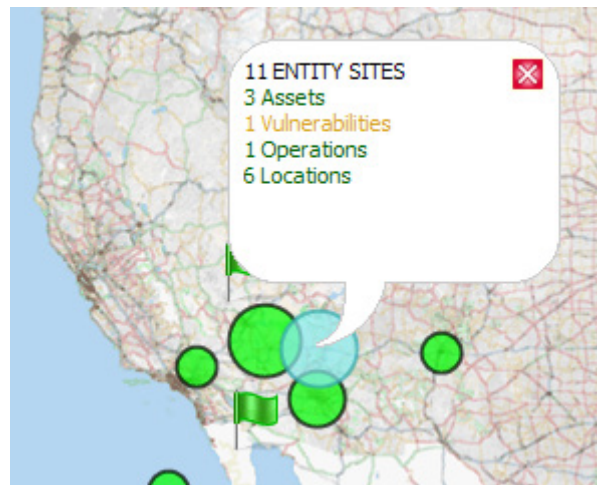


Figure 6: 2D Map Icon Aggregation.

The map itself consists of a base layer and one or more overlays. These can be selected by using the layer selector on the right hand side of the map component (see Figure 7 to see selector expanded). All of the base layers and overlays reside external to the JNDMS portal and, therefore, access to the appropriate sites must be configured.

The base layers consist of the following:

- Osmrenderer (online). This and Mapnik are both based on Open Street Maps. These provide different rendering options for this data source. The Open Street Maps provide a fairly detailed map with streets available for most communities. This is configured to access the Internet.
- Mapnik (online). This is an alternate rendering of the Open Street Maps. This is configured to access the internet.
- Metacarta (cache). This is an alternate base map. This map is very basic and tends to load quickly. This is configured to load from a local network cache (if available).
- Metacarta (online). This is the online version of the above data set.



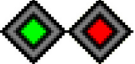

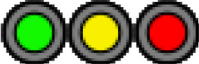



In addition to one base layer the map can show one or more overlays. The overlays available are:

- User-filtered comm. Links. This will show JNDMS communication links as associated with the locations. The links between locations is calculated based on the network infrastructure report by the enterprise tools. This will show the links between items based on the current focus and filter applied.
- User-filtered logical links. This will show the JNDMS logical links (dependencies and redundancies) between locations.
- User-filtered data. This will toggle the current focus and filtered data.
- All vulnerabilities. This will show all vulnerabilities.
- All Operations. This will show all operations.
- All Incidents. This will show all incidents.
- All Locations. This will show all locations.
- All Assets. This will show all assets.
- Osmrenderer (online). This is an overlay version of the Open Street map data. This will allow the street information to be overlaid on alternate maps.
- Mapnik (online). This is an alternate rendering of the Open Street map data.

4.3.1 Map Icons

The following table shows the icons used on the 2D and 3D maps.

Table 3: Map Icons

Function	Icon	Notes
Operation		The colour of the maple leaf identifies the current risk associated with the operation.
Location		The colour identifies risk.
Incident		The centre colour identifies if this event has been identified as an incident (red) or not (green).
Vulnerability		The annotation on the vulnerability icon identifies the risk associated with the vulnerability.
Asset		The colour of the centre identifies risk.
Network/Zone		The annotation on the zone icons shows the risk associated with the zone.
Safeguards		Safeguards are identified as a shield.
Aggregate		<p>The 2D map uses coloured circles to identify that multiple icons are in close proximity and have been aggregated into a single icon. As the map is zoomed the specific locations may become apparent.</p> <p>The 3D map makes use of Google's aggregation features. It will initially show the icons clumped together, however when the user clicks on the icon grouping they will be expanded to show the individual icons.</p>

4.4 3D Map

The 3D map in JNDMS is provided by the Google Earth Plugin. This view (see Figure 7) shows the same information that is available from the 2D map, however it provides the ability to see satellite overlays on a 3D globe. The user can zoom in and out as well as tilt the views in the same ways that Google Earth can be used.

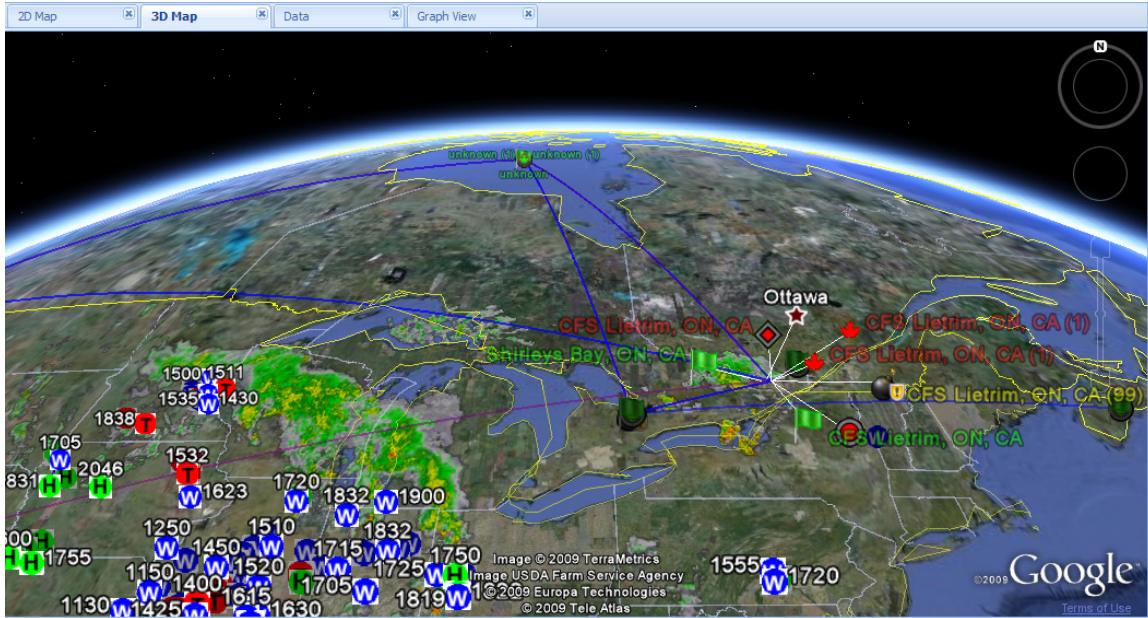


Figure 7: 3D Map

The JNDMS specific content for geographical display is provided by a servlet that creates KML data. This KML allows the same information to be sent to the 2D and 3D maps. The presentation component will just show them differently.

The Google Earth Plugin does not provide the equivalent to the layer select so an alternate configuration dialog (see Figure 8) is provided. The 'gear' icon on the top right of the primary view is used to get access to this dialog. The 3D map provides some layers from Google such as borders and roads as well as sample data from external sites such as weather warnings. The JNDMS layers are also available.

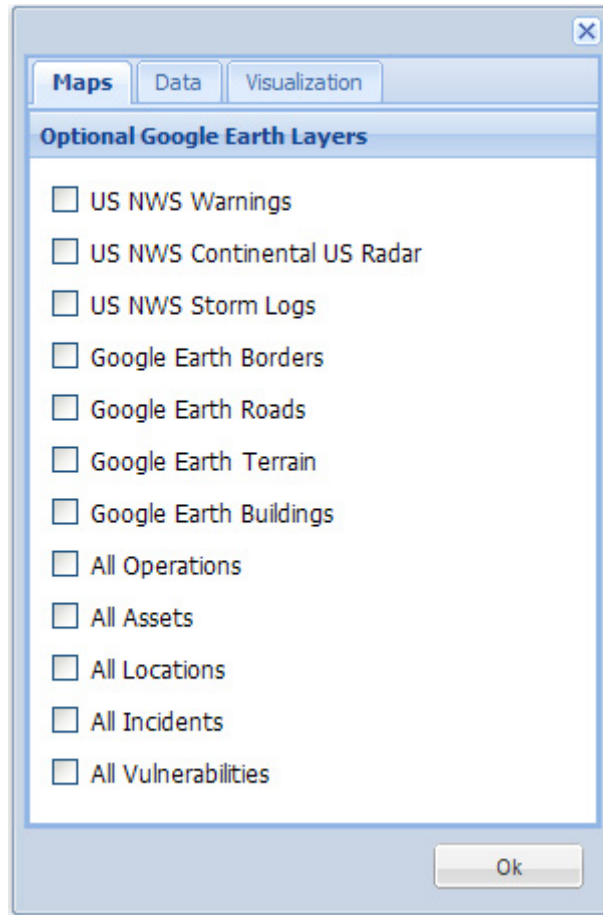


Figure 8: 3D Map Options

The icons on the map are clickable in much the same way as the 2D map. The content of the popups are contained in the KML and should be functionally the same. The links provided in the popups are clickable the same as the 2D map.

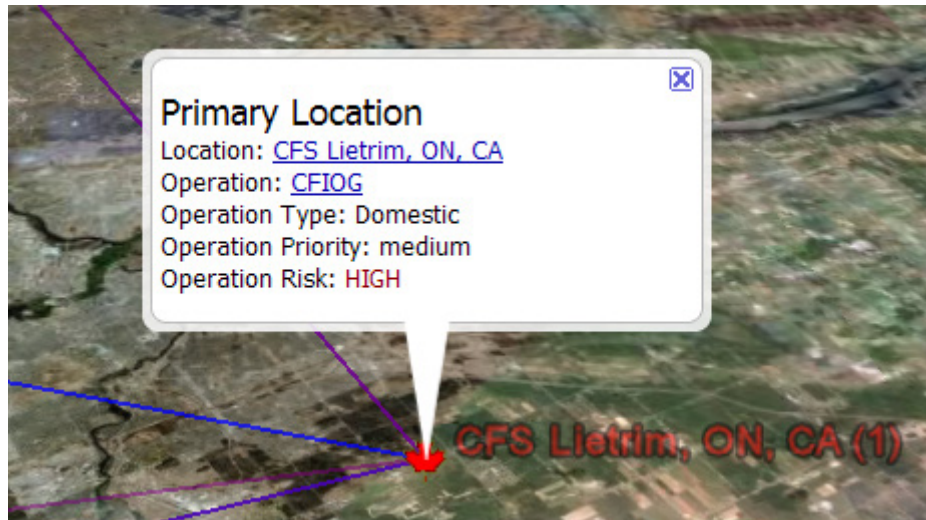


Figure 9: 3D Map Operation popup

The issue of too many icons in one spot will also impact the 3d map. In this case it is the Google Earth Plugin that provides help to separate out the cluster of icons. When a cluster of icons is selected Google Earth will expand the cluster so that the user has access to the individual icons.

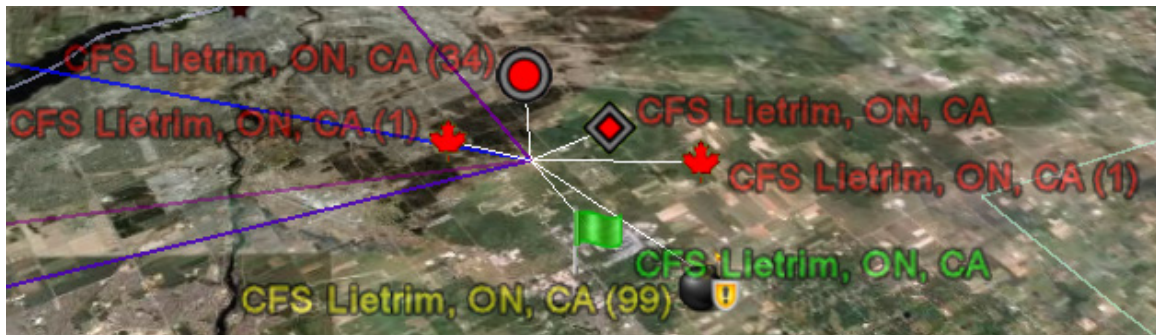


Figure 10: 3D Map Icon Cluster.

4.5 Data View

The data view provides HTML text tables and lists to explore the detailed information. Most of the navigation links will provide either a list view or a detail view.

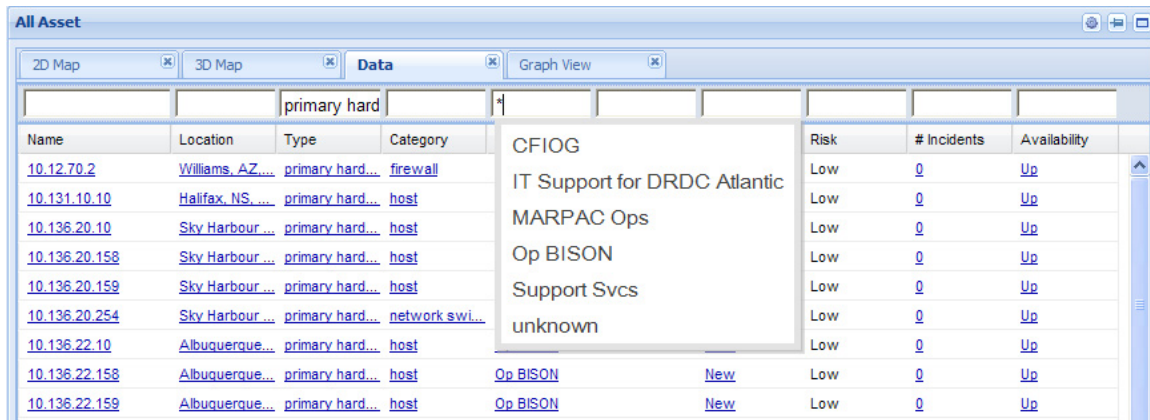
4.5.1 List Views

Each list view is generated asynchronously when the navigation request is made (by selecting an item in the navigation tree, for example). Each of the list views is generated from a common list widget that has some common features.

Each list will provide column headings, column filters, and a paging tool bar. The columns available will be based on the current focus and each column is resizable.

Much of the content within the list may be in the form of hyperlinks. When these are clicked they will send a navigation request to the other view (secondary or primary depending on which one you are clicking in). When a hyperlink is double clicked the current view will update its contents to the new focus and filter.

Above each of the column headings are a row of column filters (see Figure 11). These filters can be used to filter or reduce the amount of data presented. Each of the filter boxes is a text entry field and when 'enter' is pressed that text is used to filter that column. Each of these filters also use dynamic updates to show possible filters as you type. You can use wild cards (* and ?) in these filters. One common trick is to enter an '*' and see a list of possible filters.



Name	Location	Type	Category	Risk	# Incidents	Availability
10.12.70.2	Williams, AZ...	primary hard...	firewall	Low	0	Up
10.131.10.10	Halifax, NS...	primary hard...	host	Low	0	Up
10.136.20.10	Sky Harbour...	primary hard...	host	Low	0	Up
10.136.20.158	Sky Harbour...	primary hard...	host	Low	0	Up
10.136.20.159	Sky Harbour...	primary hard...	host	Low	0	Up
10.136.20.254	Sky Harbour...	primary hard...	network swi...	Low	0	Up
10.136.22.10	Albuquerque...	primary hard...	host	Low	0	Up
10.136.22.158	Albuquerque...	primary hard...	host	Low	0	Up
10.136.22.159	Albuquerque...	primary hard...	host	Low	0	Up

Figure 11: List View Column Filters

Each of the focus' will determine which columns are available, however you can selectively hide any of the columns. This is done by selecting the small down arrow in each column header (see Figure 12).

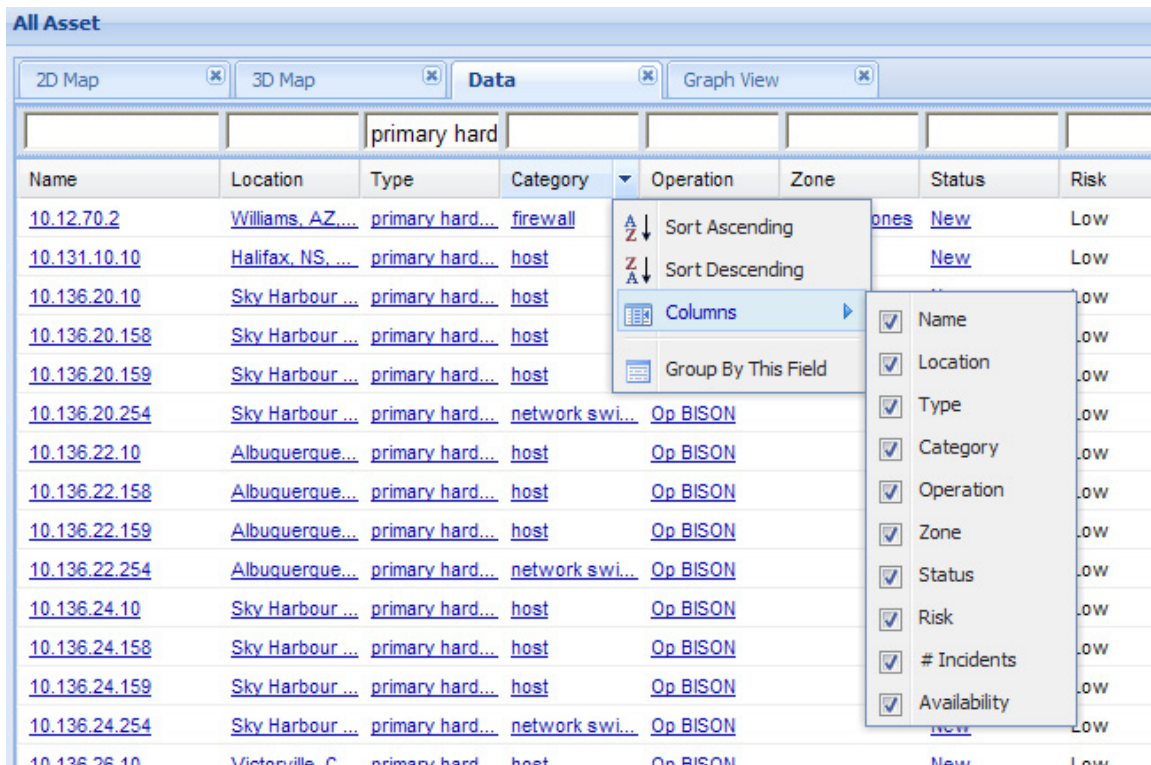


Figure 12: List View Column Selection

At the bottom of each list view is a paging tool bar. Each list will only show one page of data at a time so that the portal will not be overwhelmed with too much data. This tool bar (see Figure 13) gives the user the ability to see how many items (right hand side), how many pages (left side) and the ability to navigate through the pages. A refresh button is also provided to reload the current data.

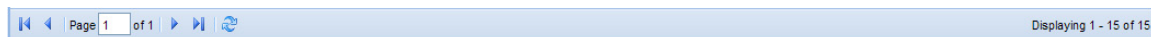


Figure 13: List View Paging Toolbar

Some of the list views are editable as well. This will be shown in the paging tool bar by the addition of an 'add' and 'remove' button (see Figure 14). These will allow new entries to be added based on the current focus.

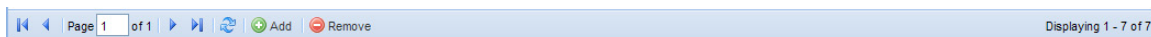


Figure 14: List View Toolbar with Edits

Some of the list views are hierarchical in nature. These views may be presented as a tree (see Figure 15) and allow the user to expand nodes.

Name	Type	Category
▶ 192.168.50.1	primary hardware	router
▶ 10.32.12.26	primary hardware	host
▶ 10.136.55.158	primary hardware	host
▶ 10.10.86.2	primary hardware	firewall
▶ 10.136.52.158	primary hardware	host
▶ 10.136.56.159	primary hardware	host
▶ 10.136.55.159	primary hardware	host
▶ DND Email	network service	service
▶ Exchange	network service	service

Figure 15: Tree List Views

The following identifies the columns available for each list view:

- Operations. This will be a list of operations
 - ♦ Name. The name of the operation.
 - ♦ Type. The type of the operation.
 - ♦ Priority. The priority (between operations) given to this operation.
 - ♦ # Sites. Sites (locations) associated with this operation.
 - ♦ # Incidents. Incidents associated with this operation.
 - ♦ Risk. The current risk score for the operation.
- Assets
 - ♦ Name. The name of the asset.
 - ♦ Location. The current location of the asset.
 - ♦ Type. The type of the asset (see section 4.1)
 - ♦ Category. The category of the asset.
 - ♦ Operation. This is the operation that has responsibility for the asset. An asset may be depended upon by more than one operation, but only one operation can be responsible for the asset. To view all related operations you must use the asset detailed view.
 - ♦ Zone. The zone the asset is in. This may list 'multiple' if the asset spans zones.

- ♦ Status. The status of the asset relates to its configuration status. This could be 'new' or 'approved'.
- ♦ Risk. This is the combined risk associated with this asset. Each asset is analyzed for its risk contribution to each operation. This value would be the highest risk that this asset represents.
- ♦ #Incidents. The number of incidents tied to this asset.
- ♦ Availability. The availability (up, down, degraded) of this asset.
- Products
 - ♦ Vendor. The product vendor.
 - ♦ Product. The product name.
 - ♦ Version. The product version.
 - ♦ Type / Category. The asset type and category for this product.
 - ♦ Status. New or approved.
 - ♦ Installations. List the number of installations of this product.
- Vulnerabilities (defaults to definitions)
 - ♦ ID. This is the ID of the vulnerability definition. This would include the source (for example CVE, BID, etc) and the associated number assigned by the given agency.
 - ♦ Modified. Date this definition was last modified.
 - ♦ Status. The status of the vulnerability.
 - ♦ Description. This is a general text description of the vulnerability.
 - ♦ # Instances. This identifies how many instances (vulnerable assets) have been currently found.
- Vulnerability Instances
 - ♦ ID. This is the ID of the vulnerability definition.
 - ♦ Source. This identifies the source that identified the vulnerability. This could be a reporting vulnerability scanner or it could have been an analysis done by JNDMS.
 - ♦ Status. This will identify if the vulnerability is 'new' or 'mitigated'.
 - ♦ Modified. The last time an update was received for this instance.
 - ♦ Asset Name. The name of the asset that is vulnerable.
 - ♦ Asset Category. The category of the asset that is vulnerable.
- Exploits. Exploits are components of vulnerabilities in the JNDMS model. These identify one or more methods that the vulnerability could be exploited. Many sources don't separate the exploit from the vulnerability so for these cases a 'default' exploit is made for each vulnerability.
 - ♦ VulnID. This is the ID of the associated vulnerability.

- ♦ ID. An identifier that is unique for the given associated vulnerability. Many exploits will have the ID 'default' to ensure that at least one exploit is tracked.
- ♦ Description. This is the text description for the exploit.
- ♦ Availability. This identifies if the exploit is unproven or widely available (based on CVSS scoring).
- ♦ Date. The date this exploit was identified.
- ♦ Access Vector. This identifies if local access is required or if it can spread over the network.
- ♦ Authentication. This identifies if authentication is required for this exploit to be effective.
- ♦ Popularity. How popular this exploit is.
- ♦ CVSS Score. The CVSS score (from CVE entries).
- Events
 - ♦ ID. This is a unique identifier for the event.
 - ♦ Root Event. This is a flag to identify if this event is a root event. In JNDMS a single event may be related to other events in a cause / effect relationship. The event that is the cause is the parent of those that are the effects.
 - ♦ Type. This is the type of event, such as compromise or policy violation.
 - ♦ Status. This is the status of the event such as active, resolved or mitigated.
 - ♦ Location. This is the location associated with the impact of the event, if any.
 - ♦ Asset. This is the asset associated with the event.
 - ♦ Created. The date this event was created.
 - ♦ SP* (DSS Priority). This is a priority calculated by the analysis to help identify events causing issues.
 - ♦ Severity. This is the severity of the event based on the importance of the assets that were impacted.
- Safeguards
 - ♦ ID. This is a unique identifier.
 - ♦ Type. This is the type of the safeguard such as firewall or patch.
 - ♦ Description. This is a textual description of the safeguard.
 - ♦ Data Source. This is the source that reported this safeguard.
 - ♦ Efficiency C/I/A. This identifies how effective this safeguard is in protecting Confidentiality, Integrity and Availability. This allows for cases where safeguards are only partially effective.
 - ♦ Policy ID. This gives a link to potential policies. This is for reference only.
 - ♦ SensorSigID. This gives a potential link to signatures that are tracked.

- ♦ Modified. The last modified date.
- Locations
 - ♦ Name. Name of the location.
 - ♦ Description. Textual description.
 - ♦ Latitude / Longitude. Position information.
- Network
 - ♦ Zone. This is the network zone
 - ♦ Risk. Risk associated with this network. Network risk is based on the assets within the zone.
 - ♦ Probability of Attack. This is a computed value based on the events and vulnerabilities within the zone.
 - ♦ #Subnets. The number of subnets that make up this network.
 - ♦ #Safeguards. The number of safeguards within the zone.
- Points of Contact. The following fields refer to information about each point of contact.
 - ♦ Rank.
 - ♦ Name
 - ♦ Position
 - ♦ Function
 - ♦ CSN Phone
- RFC. The following fields refer to information about the RFC.
 - ♦ ID. This is a unique identifier.
 - ♦ Status. This identifies if the RFC is new, is in progress or if it has been completed.
 - ♦ Start Date. This is the planned or actual start date of the RFC.
 - ♦ Completion Date. This is the planned or actual completion date of the RFC.

4.5.2 Detailed Views

Each detailed view has two components (see Figure 16). The panel on the left (1) is a list box that has the name of each of the content panels. These may represent a collection or common information or a relationship to other entities. When one of the entries in the list box (1) is selected, the content pane (2) will show the associated information.

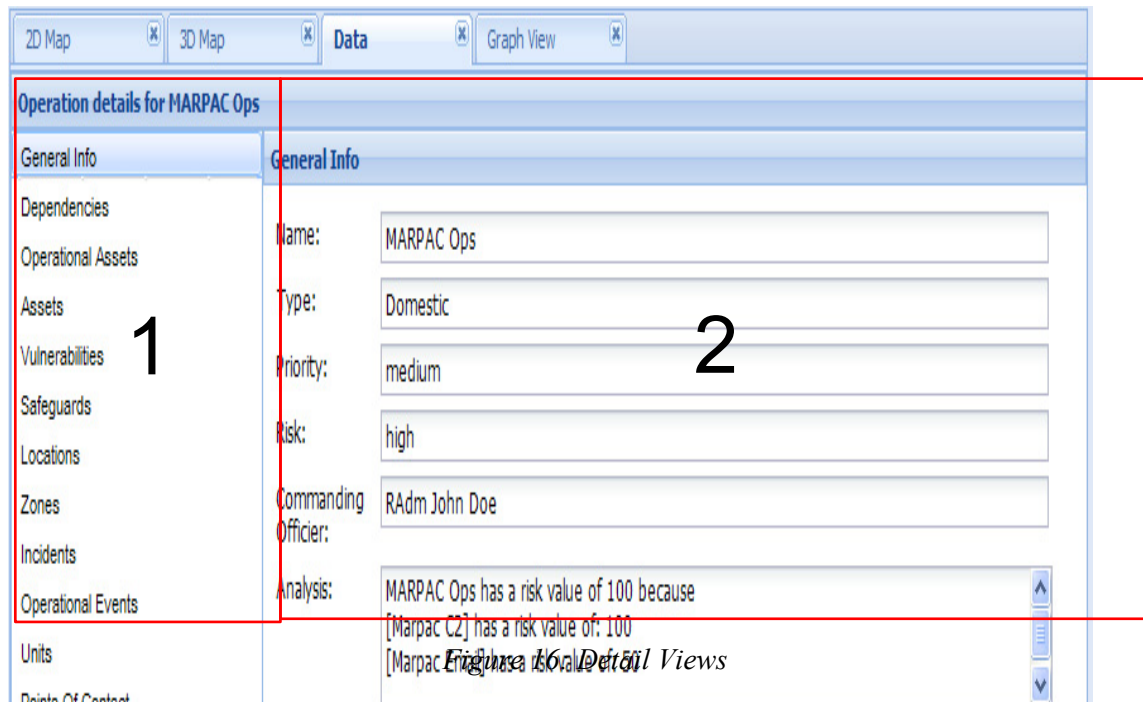


Figure 16: Detailed Views

The following identifies the content available (as shown in the list box, #1 above) for each of the core entities.

- Operation
 - ♦ General Information.
 - Name of the operation.
 - Type of the operation, such as domestic or deployed.
 - Priority. This is a value to compare different operations.
 - Command officer.
 - Risk. This is based on the combined risk associated with any asset required by this operation.
 - Analysis. This provides a summary of the risk analysis done for this operation.
 - ♦ Dependencies. This provides a tree view of the operational dependencies.
 - ♦ Operational Assets. This provides a flat list of all of the assets that this operation depends on. This view of the assets shows the relationship to this operation.

- ♦ Assets. This shows the dependent assets with the columns found in the generic asset lists.
- ♦ Vulnerabilities. This shows all vulnerabilities associated with this operation.
- ♦ Safeguards. This shows all safeguards associated with this operation.
- ♦ Locations. This shows all locations associated with this operation.
- ♦ Zones. This shows all zones associated with this operation.
- ♦ Incidents. This shows all incidents associated with this operation.
- ♦ Operational Events. This shows operational events associated with this operation. An operational event gives windows of time for the expected activity of the operation. Most operation will have a default event that spans the life time of the operation.
- ♦ Units. This lists who (what units) comprise this operation.
- ♦ Points of Contact. This identifies points of contact for this operation.
- Assets
 - ♦ General Information
 - Category / Type. See section 4.1.
 - Status. This identifies if this is a new asset or part of an approved configuration.
 - Enabled. This identifies if this asset is enabled. An asset can be disabled if it is known to be in storage or transit.
 - Created/Modified. Time stamps for this asset.
 - IP Address. The IP address of this asset. This may be the default IP address for a host (if it has more than one IP address) or it may be the IP address of the network interface card (NIC).
 - Vendor/Product/Version. If this is a known product the information would be found here.
 - Importance. This is a rating identified by the JNDMS analysis based on how this asset is used.
 - Heart rate. Possible identification of how often this asset reports or is scanned (generally for availability).
 - Latitude/Longitude. Position information.
 - Network host. If this is a peripheral or network card this will identify the associated host.
 - Active incidents. This shows active incidents associated with this asset.
 - Risk override. This will identify if risk override is in effect. The user can override the calculated value of risk if they feel it is not appropriate. At any point the risk override can be disabled and the automatic risk value would be used.

- Risk. This is the current risk value. This could be either a user entered value if the risk override is enabled or the DSS calculated risk score.
- Availability. This is the current availability of this asset. This could be either up, down or degraded.
- Asset Count. This asset could represent more than one asset. This would show how many identical assets this represents.
- Analysis. This provides a textual description of the risk analysis.
- ♦ Asset vulnerabilities. This shows all vulnerabilities associated with this asset.
- ♦ Vulnerability ports. This shows the ports that have vulnerabilities currently associated with them. This information generally depends on a vulnerability scanner providing port details.
- ♦ Safeguards implemented by Asset. This identifies the case where this asset is a safeguard.
- ♦ Assets protected by this asset. If this asset is a safeguard this will show the assets being protected.
- ♦ Assets that protect this asset. This will show safeguard assets that are protecting this asset.
- ♦ Asset that protect this asset through the zone. This will show perimeter safeguards that may be protecting this asset. The protection of perimeter safeguards depends on the path taken through the network.
- ♦ Asset availability. This will identify the availability of this asset.
- ♦ Operations. This identifies the associated operations.
- ♦ Asset locations. This identifies the associated locations.
- ♦ Zone containing asset. This identifies the zone or zones that contain this asset. An asset may exist in more than one zone if it is multi-homed.
- ♦ Incidents involving asset. This identifies incidents that have a relationship to this asset.
- ♦ Asset required by this asset
- ♦ Assets requiring this asset
- ♦ Redundant assets
- ♦ Communication links
- ♦ Points of contact
- ♦ RFC's.
- ♦ Product
- ♦ Change log

- Vulnerabilities
 - ♦ General info
 - ID. Each vulnerability must have a unique identifier. This will include the type of the identifier as well as a secondary identifier. The type is based on the agency or source responsible for the vulnerability and the secondary identifier is the unique identifier assigned by that agency. For example the vulnerability might be based on the Common Vulnerability Enumeration (CVE). The vulnerability type would be 'CVE' in this case and the secondary identifier would be the ID assigned by CVE.
 - Category. The category of the vulnerability identifies if this is a cyber (software) or physical vulnerability.
 - Description. This is a textual description of the vulnerability.
 - Created/Modified. These timestamps identify when the vulnerability was created and when it was last modified.
 - Source. This identifies the source of the vulnerability. The source may be a particular database or tool that is reporting this vulnerability.
 - Type. The type of the vulnerability identifies a broad class of how this vulnerability impacts its target. Typical values would identify either cyber types such as application, database or operating system, or it may identify physical types such as weather or power outages.
 - Security Level. This identifies the security level of the vulnerability.
 - Status. This identifies the JNDMS view of the status of the vulnerability. This will be one of new, approved, pending or denied.
 - Number of Products. This identifies how many products are associated with the vulnerability.
 - ♦ Exploits. This relationship identifies the exploits associated with the vulnerability. Many sources do not separately track exploits and, therefore, each vulnerability may have a 'default' exploit created.
 - ♦ Products. This relationship identifies the list of products that are vulnerability to this vulnerability.
 - ♦ Assets. This list shows the assets that have this vulnerability. This relationship is also known as the vulnerability instances.
 - ♦ Safeguards. This list shows the safeguards that have been identified to have some mitigating impact on this vulnerability.
 - ♦ Incidents. This list shows the incidents that are related to this vulnerability.
 - ♦ RFCs. This identifies the RFCs that are related to this vulnerability. For example an RFC may identify a patch that will mitigate this vulnerability.
 - ♦ Change Log. This will identify significant changes to this vulnerability definition record.

- Events
 - ♦ General Info
 - Description. This is a textual description of the event.
 - Incident ID. This is a unique identifier for this event or incident.
 - Created/Modified. These are timestamps related to when we first knew about the event and when we last had an update. These are related to when the event was reported to JNDMS.
 - Event time. This is the timestamp related to the time of the event. This may differ from the above created timestamp if the reporting agent or source could provide additional information.
 - Is Incident flag. If this event had been identified as an incident this flag will be set to 'Y'.
 - Type. This is the type of the incident. This can include security or infrastructure.
 - Disposition. This helps to classify the incident and may include values such as threat, violation, safeguarded, etc.
 - Confidence. This will identify the level of assessment that has been completed on this incident or event. This may initially be set to 'detection' then updated based on the analysis level.
 - Status. This field identifies the status of the incident such as active, mitigated, resolved or forecasted.
 - Parent Incident. Incidents in JNDMS can be hierarchical in nature if a cause and effect relationship has been identified. This will identify the parent of this incident.
 - Notes. This is a general notes field.
 - Alert. This is a text field to identify alerts.
 - System Events. This is a field in which system events from the source can be viewed.
 - Operational dependency value. This is a score that relates how this incident has impacted operational requirements. This should be viewed as a relative score compared to other incidents.
 - Source priority. This is the priority that was assigned by the reporting source.
 - Analysis priority. This is the priority that has been assigned by the JNDMS analysis.
 - Security level. This is the security level of this incident. It is generally initially set by the security level of the originating network or agent, however can be increased later.
 - Logs. This provides a place to store logs, or partial logs, associated with this incident.

- Formatted reports. This provides a place to store reports associated with this incident.
- Location. This is the location that identifies the possible impact or target of this event.
- Ticket ID. This provides a place to store a related ticket ID if there is one.
- Ticket Status. If this event is related to a ticketing system then this field can provide the status of the originating ticket.
- Severity. This should identify a normalized value from 0-100 that identifies the severity of impact of this incident.
- Data Source. This field identifies the agent or system that provided this event to JNDMS.
- ♦ Incident Sensor Path. This identifies, when available, the IP address, the asset name, the receive time, the completion time, the event type, the severity and the priority of this event. This chart identifies each of these values with respect to the source, the sensor, the target and the JNDMS analysis.
- ♦ Impact
 - Success probability. This is the score, based on the JNDMS analysis, that this event will result in a successful exploitation of the system.
 - Is Incident. This identifies if this is considered an incident or just an event. The deciding factor is if we can identify an impact based on the event.
 - Vulnerability ID Type. If this event is related to a vulnerability this will identify the ID type of the vulnerability.
 - Affected Asset. If an asset could be identified as either impacted by this event or the target of this event it will be identified as the affected asset.
 - Priority. The priority is a score assigned to the incident based on the analysis. This should be viewed relative to the priority scores of other incidents.
 - Alert. This is a text field for related alerts.
 - Environmental Damage Values and Base score. These values may be computed if this incident is the result of a vulnerability.
 - Impact (C/I/A). This identifies the potential impact to C, I and A to assets considered affected by this incident. These scores are originally based on the incident type, however can be modified for individual incidents.
 - Severity. This is an assessment of the severity of this incident that is based on the impact assessment. This is a normalized score from 0 – 100.
- ♦ Child Incidents (tree view). Incidents in JNDMS can be hierarchical in nature if we have created or have been able to discern a cause and effect relationship. This view will show this relationship.
- ♦ Assets. This will identify assets associated with this incident, including possible sources or sensors.

- ♦ Affected Assets. This relationship identifies assets that have been impacted or targeted as a result of this event or incident.
- ♦ Source of the incident. This identifies the source of the incident and would be interpreted based on the type of incident. The source should be considered the cause of the incident and may represent the host spreading a virus or the source of an attack.
- ♦ Location of the sensor. This identifies the location of the sensor that identified this event and is reporting to JNDMS.
- ♦ Affected Operations. This is a list of operations that are potentially impacted or threatened by this event.
- ♦ Associated Vulnerabilities. This identifies any vulnerabilities that may be associated with this incident.
- ♦ Correlated Incidents. There are a number of factors that are examined to try and determine if there are any shared attributes for multiple incidents. These are analyzed and a weighted correlation is identified on how closely these incidents are related. This list shows other incidents that are correlated (i.e. share attributes) with the current incident.
- ♦ Correlation Details. This field identifies the reasons for the correlations.
- ♦ Zones. This lists the zones related to this incident.
- ♦ SOPs. This lists potentially related SOPs (Standard Operating Procedures) related to this incident.
- ♦ History. This identifies changes to this incident over time.
- Safeguards
 - ♦ General info
 - ID. This is a unique identifier for this safeguard.
 - Data Source. This identifies what agent or system reported this safeguard.
 - Efficiency [C/I/A]. These values identify how effective this safeguard is with respect to C, I and A.
 - Policy ID. This field identifies if this safeguard is related to a specified policy.
 - Sensor Sig ID. This field identifies if there is an associated signature that relates to this safeguard.
 - Created/Modified. These are timestamps related to when we knew about the safeguard and when it was last updated.
 - Description. This is a textual description for this safeguard.
 - Type. This identifies the type of this safeguard such as if it is a firewall, a virus scanner, a patch or another type.
 - ♦ Assets implementing. This is a list of all assets that can provide this safeguard.

- ♦ Assets protected by. This relationship identifies the assets that are protected by this safeguard.
- ♦ Safeguarded vulnerabilities. This list identifies which vulnerabilities are mitigated to some degree by this safeguard.
- ♦ Zones bordered. Safeguards, in JNDMS, can be two general types. The first is a targeted safeguard that protects specific assets from specific vulnerabilities. The other type is a border safeguard that protects assets within a network zone depending on the access vector of the threat. This field identifies any borders between zones that are protected by this safeguard. This would only be available for the border type safeguards.
- ♦ Zone rules. Border safeguards may have one or more rules associated with them to perform their function. Firewalls are the most obvious example of this type of safeguard. This table lists the rules associated with this safeguard.
- ♦ RFCs. This table will list the RFCs that are associated with this safeguard.
- Locations
 - ♦ General Info
 - Assets implementing. This is a list of all assets that can provide this safeguard.
 - Assets protected by. This relationship identifies the assets that are protected by this safeguard.
 - Safeguarded vulnerabilities. This list identifies which vulnerabilities are mitigated to some degree by this safeguard.
 - Zones bordered. Safeguards, in JNDMS, can be two general types. The first is a targeted safeguard that protects specific assets from specific vulnerabilities. The other type is a border safeguard that protects assets within a network zone depending on the access vector of the threat. This field identifies any borders between zones that are protected by this safeguard. This would only be available for the border type safeguards.
 - Zone rules. Border safeguards may have one or more rules associated with them to perform their function. Firewalls are the most obvious example of this type of safeguard. This table lists the rules associated with this safeguard.
 - RFCs. This table will list the RFCs that are associated with this safeguard.
 - ♦ Assets. This lists the assets found at this location.
 - ♦ Vulnerabilities. This lists the vulnerabilities found at this location. This list is the vulnerability definitions.
 - ♦ Vulnerability instances. This lists the individual vulnerable assets and their related vulnerabilities at this location.
 - ♦ Operations. This lists the operations related to this location.
 - ♦ Safeguards deployed. This identifies what safeguards have been deployed at this location.

- ♦ Zones. This identifies which network zones have a presence at this location.
- ♦ Incidents. This lists the incidents related to this location.
- Network. The networks within JNDMS are identified as unique zones.
 - ♦ General info
 - Name. This is the name of the network zone.
 - Description. This is a textual description of the network or zone.
 - Zone ID. This is a unique identifier for the zone.
 - Probability of Attack. This is the current assessment of the possibility of an attack originating from this zone. This is used to determine overall risk scores.
 - Latent probability of attack. This is a base score used as a starting point for the probability of attack. This value is generally based on historical behaviour of a network or an operator's assessment. Activity such as vulnerabilities or attacks are used to modify this score and calculate the probability of attack.
 - Risk. This is the current risk score associated with this network based on all of the assets within the zone and their relationships to operations.
 - Created / Modified. These are timestamps used to track updates to our knowledge of this network.
 - Notes. These are general purpose notes.
 - ♦ Assets protecting zone. This list identifies assets that protect this zone. An asset is identified to be protecting the zone if it implements a safeguard that protects an asset within the zone.
 - ♦ Assets in zone. This list identifies all assets in this zone.
 - ♦ Vulnerabilities. This list identifies all vulnerabilities that have a presence within the zone.
 - ♦ Safeguards. This list identifies all safeguards implemented within this zone.
 - ♦ Operations affected. This list identifies all operations related to assets within this zone.
 - ♦ Locations affected. This list shows all locations in which this zone has a presence.
 - ♦ Adjacent zones. This list identifies directly accessible adjacent zones.
 - ♦ Incidents. This lists all incidents related to this zone.
 - ♦ RFCs. This lists all RFCs that target this zone.
 - ♦ Zone rules. This will show a list of rules associated with border safeguards of this zone.
 - ♦ Subnets. This identifies the network subnets that comprise this zone.
 - ♦ Points of contact. This will list all points of contact associated with this zone, including assets within the zone.

4.6 Graph View

The graph view provides an interactive method for exploring some of the same entities and relationships found in the other views. The graph view is comprised of an applet within the portal (see Figure 17).

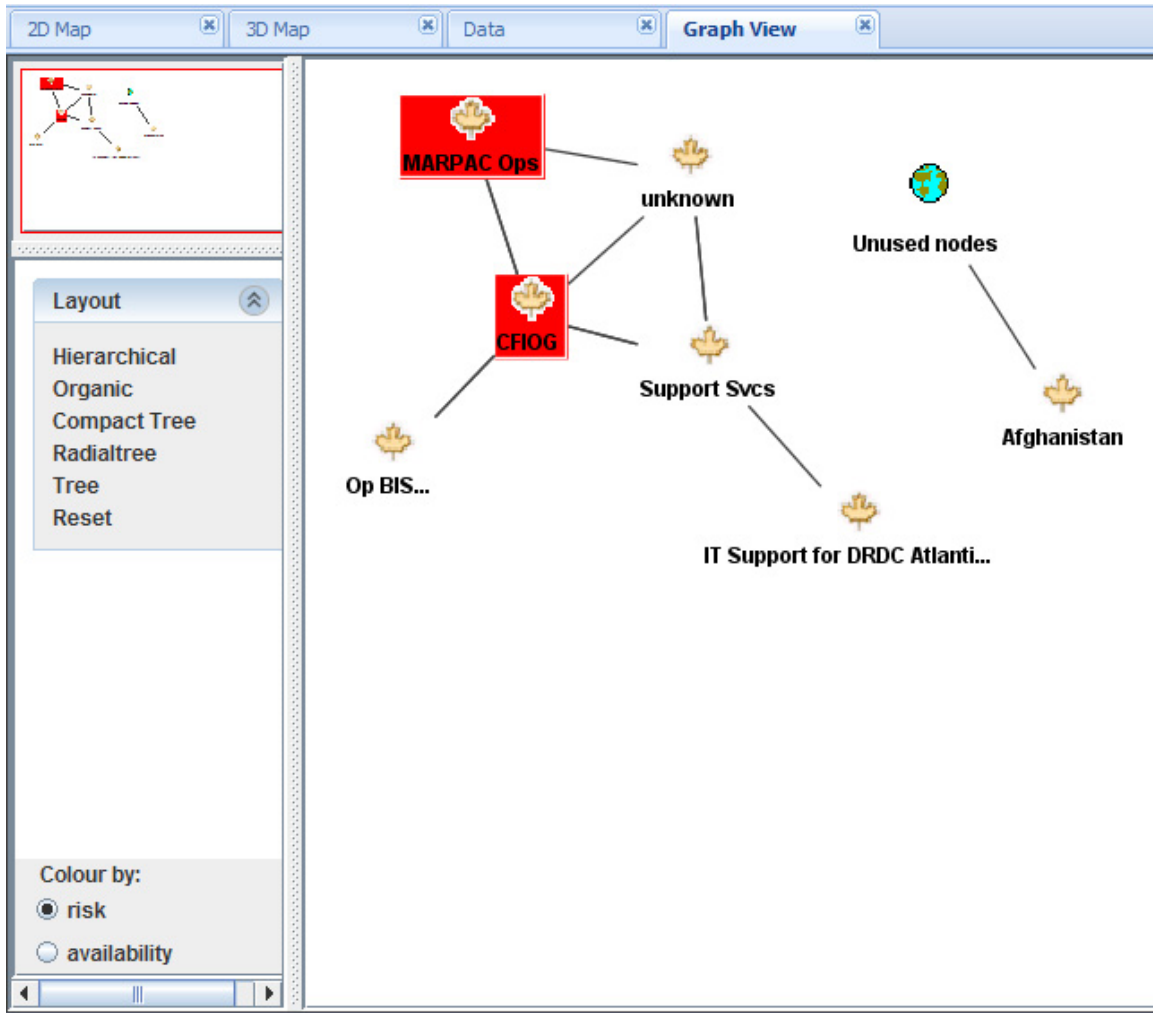


Figure 17: Graph View Applet

Each icon within the applet represents one of the entities previously identified (see section 4.1). When the icons are connected there is a relationship between the two items. The applet also provides the ability to choose alternate layout algorithms and to colour the icons either by risk or availability.

When the applet is first started, or when a new navigation item is selected, the applet must load the initial nodes or roots of the graph that it will display (see Figure 18). When this occurs a message saying ‘Loading Roots’ will be displayed (see figure below). This message contains an ‘X’ that provides the ability to cancel this operation if required.

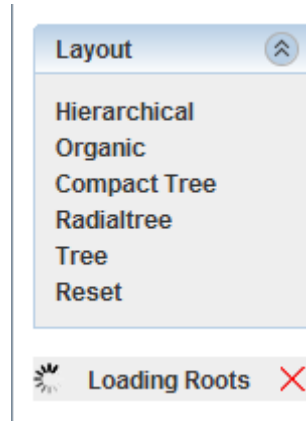


Figure 18: Graph View Loading Initial Roots

A right click menu is available for each of the icons (see Figure 19). This menu gives the ability to manipulate the view in the windows (zoom, fit, set actual size), the ability to show or hide related icons and the ability to view the current item. When ‘view item’ is selected a navigation event is sent to the secondary view which will display the detailed information.

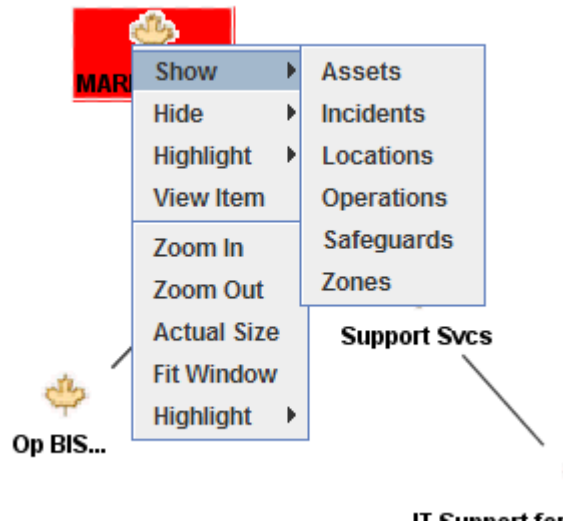


Figure 19: Graph View Entity Expansion

The ‘show’ option allows the user to selectively navigate the relationships between the entities. The user can show related assets, incidents, locations, operation, safeguards and zones. In each case the new nodes are added to the visible graph and any associations (lines) to existing nodes are added.

The lines in the graph identify a general relationship between the two nodes. The relationships can be general, dependent or redundant. See Figure 20 for an example.

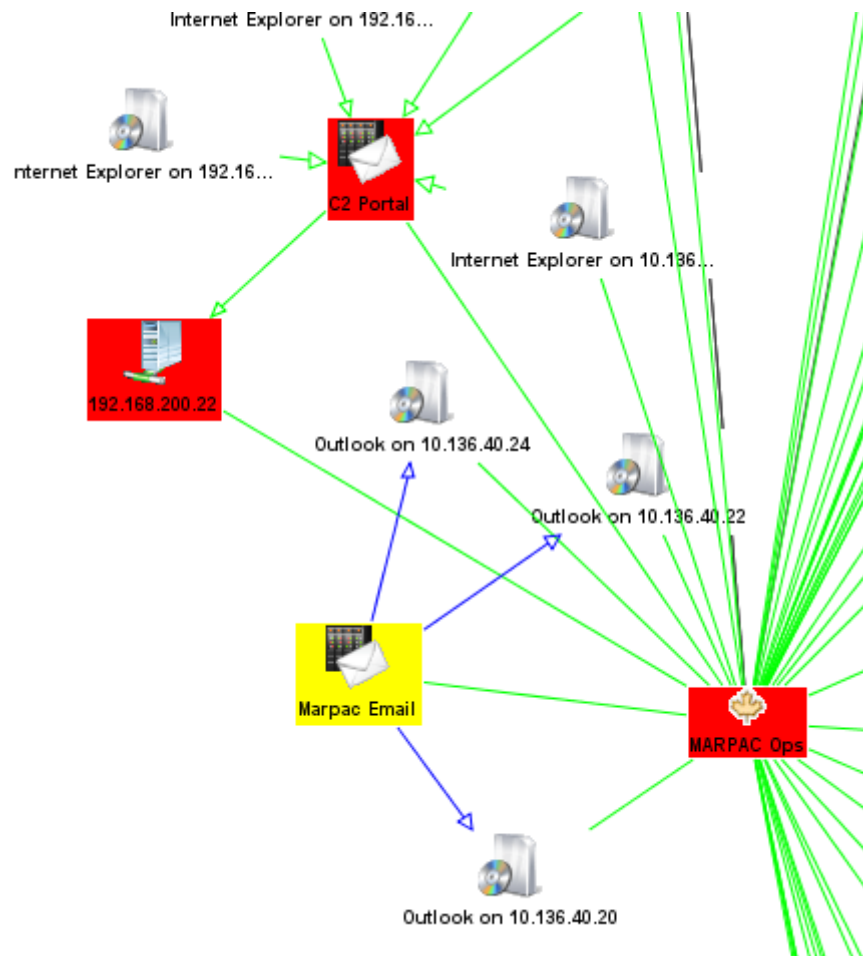


Figure 20: Graph View with Dependent and Redundant Links.

The black lines represent a generic relationship. The blue lines represent a redundant relationship and the green lines represent a dependent relationship. Some of the relationships will identify the direction through an arrow head.

The applet also has the ability to view certain aspects of the network over time. At the bottom of the applet is a time field in which an alternate time in the past can be entered. This will update the current view to show the status of the visible assets as they were at that point in time.

4.7 Search and Filters

The portal provides the ability to use search and filters to understand the data. Both the search and the filter are activated using the 'search' and 'filter' panels on the main side bar.

The filter provides the ability to set criteria that will be applied to all views and all queries. The current portal only offers a few simple filters however, when active, this will impact all views. The 'filter' pane in the side bar will identify if the filter is enabled or disabled.

The search ability provides a dialog box that the user can use to define a custom search. This custom search may return records of any entity type. The resulting view is very similar to the detailed views used. The content types listed will be one for each entity type.

The search offers a text search (see Figure 21) in which the user can type any text and select which fields will be searched.

Text Search

Search Text:

Find Results: ☒ any word ☐ all words ☐ exact phrase

Text Fields to Search

CIRT fields	NVAT fields	J6 fields	Points of Contact	Change Log fields
<input checked="" type="checkbox"/> Incident Description	<input checked="" type="checkbox"/> Vulnerability ID	<input checked="" type="checkbox"/> Operation Name	<input checked="" type="checkbox"/> First Name	<input checked="" type="checkbox"/> Key
<input checked="" type="checkbox"/> Incident Type	<input checked="" type="checkbox"/> Vulnerability Description	<input checked="" type="checkbox"/> Operation Priority	<input checked="" type="checkbox"/> Last Name	<input checked="" type="checkbox"/> Table
<input checked="" type="checkbox"/> Incident Status	<input checked="" type="checkbox"/> Product Name	<input checked="" type="checkbox"/> Operation Type	<input checked="" type="checkbox"/> Function	<input checked="" type="checkbox"/> User
<input checked="" type="checkbox"/> Incident Location	<input checked="" type="checkbox"/> Product Vendor	<input checked="" type="checkbox"/> CO Name	<input checked="" type="checkbox"/> Rank	<input checked="" type="checkbox"/> Field
		<input checked="" type="checkbox"/> Operation Notes	<input checked="" type="checkbox"/> Position	<input checked="" type="checkbox"/> New Value
		<input checked="" type="checkbox"/> Location Name	<input checked="" type="checkbox"/> Organization	
		<input checked="" type="checkbox"/> Zone Name	<input checked="" type="checkbox"/> Location Name	
		<input checked="" type="checkbox"/> Zone Description	<input checked="" type="checkbox"/> Email Address	
		<input checked="" type="checkbox"/> Asset Name	<input checked="" type="checkbox"/> Comments	
		<input checked="" type="checkbox"/> Asset Description	<input checked="" type="checkbox"/> Work Phone	
		<input checked="" type="checkbox"/> Asset Approval Status	<input checked="" type="checkbox"/> Cell Phone	
			<input checked="" type="checkbox"/> Home Phone	

Figure 21: Text Search

As part of the same search dialog the user can also select a date search. This provides the ability to select the start and end date as well as the date fields that will be searched (see Figure 22).

Date Search

Lower:

Upper:

Date Fields to Search

CIRT fields	NVAT fields	J6 fields	Points of Contact	Change Log fields
<input checked="" type="checkbox"/> Incident Opened Time	<input checked="" type="checkbox"/> Vulnerability Opened Time	<input checked="" type="checkbox"/> Operation Creation Date	<input checked="" type="checkbox"/> Created Date	<input checked="" type="checkbox"/> Change Date
<input checked="" type="checkbox"/> Incident Detected Time	<input checked="" type="checkbox"/> Vulnerability Released Time	<input checked="" type="checkbox"/> Operation Start Date	<input checked="" type="checkbox"/> Modified Date	
		<input checked="" type="checkbox"/> Operation End Date		
		<input checked="" type="checkbox"/> Location Creation Date		
		<input checked="" type="checkbox"/> Zone Creation Date		
		<input checked="" type="checkbox"/> Asset Creation Date		

Figure 22: Date Search

The final part of the search dialog is a search based on IP addresses (see Figure 23).

IP Search

IP Address:

☒ and ☐ or

Port:

☐ Search IP Range

IP and Port Fields to Search

CIRT fields	NVAT fields	J6 fields	AnalystDB fields
<input checked="" type="checkbox"/> Incident Target IP	<input checked="" type="checkbox"/> Vulnerability Description	<input checked="" type="checkbox"/> Zone Start IP	<input checked="" type="checkbox"/> IP Address
<input checked="" type="checkbox"/> Incident Source IP		<input checked="" type="checkbox"/> Zone End IP	<input checked="" type="checkbox"/> Port
<input checked="" type="checkbox"/> Incident Target Port		<input checked="" type="checkbox"/> Asset IP Address	

Figure 23: Search by IP Address

5 Administration and Trouble Shooting

This section identifies some common administrative and trouble shooting procedures.

5.1 System Start Up and Shut Down

There are three core components required for JNDMS to function. These components would be the portal (JUI), the core services (JSS, which includes the DSS) and the database (JDW).

5.1.1 Oracle Database

The database is an Oracle database and is configured in most environments to automatically start and load its databases when the system starts. JNDMS has been tested with both 10g and 11g, however after the DREnet deployment efforts only 11g is supported in many of the build and deployment scripts.

On most JNDMS installations the common Oracle tools should be used to manually manage the database if the automatic scripts are not sufficient. Oracle provides the commands dbstart and dbshut to start and stop the database. Oracle also provides a web based tool called the Enterprise Manager (see section below on database maintenance) that may be used to manage the database.

Some installations of Oracle, such as the DREnet deployed version, make use of a completely encrypted data store. When this feature is enabled the database cannot be automatically started. The following procedure must be used:

```
- Log in as the oracle user
- Start sqlplus: sqlplus / as sysdba
SQL> startup mount;
SQL> alter system set encryption wallet open authenticated by
SQL> "EncryptionPassword"; alter database open;
```

5.1.2 Web Applications

The other two core components (i.e. the portal and the core services) are both packaged as web applications. JNDMS is bundled with a Liferay portal which includes a web application server. The web applications are deployed and managed using this web application.

The default installation location of Liferay and the web application server is:

```
C:\jndms\liferay
```

To start the web applications run:

```
startup.bat in c:\jndms\liferay\bin
```

To stop the web applications run:

```
shutdown.bat in c:\jndms\liferay\bin
```

5.2 System Issue Diagnosis

There are occasions when the system does not respond as expected so this section will identify some methods to ensure the core components are running and some common issues that could be addressed.

5.2.1 Portal

The most straight forward method to determine if the portal is running is to log into the portal. This method is quick and allows you to select each of the view tabs to ensure that each can display their content.

The portal is a web application (JndmsPortal.war) and depends on the web application server running. For additional checks see the section below for the web application server.

5.2.2 Core Services

The core services is another web application called JSS.war. The most direct method of checking this component is to use we web browser and examine the service. The following should be available:

- [http://\[server\]/JSS](http://[server]/JSS): This page should provide some basic version information about the JSS. If this page is not available then you should examine the web application server (see section below) and the database (see below).
- [http://\[server\]/JSS/services](http://[server]/JSS/services): this page should be generated by the web services layer and show what services are available.

Another check that can be performed is through the use of the JSS client application (see section 3.5.1). This program will send information to the JSS or query the JSS. Any query on the JSS, such as looking at its current event queues will test the basic execution of the JSS.

The core services depend on the execution of the database as well as the web application server so the sections below should also be consulted. The JSS also contains the DSS and is responsible for the analysis done within the system. As such this application can be quite complex and so can the issues involved. Any issues relating to the JSS in which the JSS seems to be running should be diagnosed through the web application server logs (see below). These logs should contain detailed information on the errors.

5.2.3 Database

Any issues with the database will be apparent through both the portal and the core services. There are, however, many tools to examine the database directly, although these are slightly more advanced methods.

To determine the current credentials used by either the core services or the portal you should consult the configuration files for these applications. They are:

- Core Services (JSS): WEB-INF/jss.properties. Database properties are found as jss.db_driver, jss.db_url, jss.db_user and jss.db_pass.
- Portal: WEB-INF/web.xml. Database properties are found in the context parameters section and include the databaseConnection, databaseName and databasePassword.

The above configuration files are text and human readable. These credentials can be used in tools such as Oracle SQLDeveloper or SQLPlus. The use of these tools will confirm access to the database and can allow some quick queries to be run. For example:

- Select count(*) from asset: This, or any count on the core tables, will ensure that the schema has been correctly set up and that there are assets available. Some tables such as 'assetcategory' are lookup tables and should be pre-loaded with data. They should never be empty.

For additional information on the database see the section below on database maintenance.

5.2.4 Web Application Server

The web application server is responsible for the execution of the portal and the core services. If either of these applications is running (see sections above) then the web application server is running.

The web application server keeps a number of logs in the c:\jndms\liferay\logs directory. The jndms.log should be consulted when issues arise. A common practice would be to ensure that the application server is shutdown, delete or archive the current logs, then start the server again. This ensures that only the current issues are in the log.

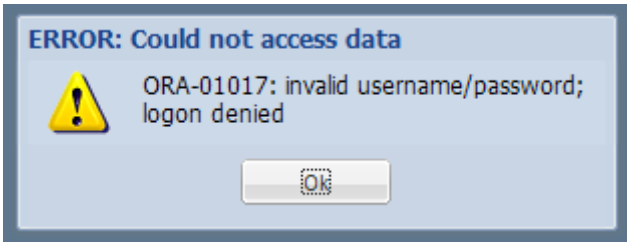
The log files can be quite verbose and can be configured with the file c:\jndms\liferay\lib\log4j.xml. There are a number of log entries that you should look for, including:


- There should be an entry that notes that the web application server has successfully started. It will include the text "Catalina.start ... Server startup in xx ms"
- There should be an entry that shows that the JSS has started initialization which should include the text "JSSConfig ... JNDMS ID: x".
- Any entries at the ERROR level or logging of Java exceptions should be examined more closely.

5.2.5 Common Issues

This section will identify a number of common issues, their symptoms and possible solutions.

Table 4: Common Issues and Solution

Issue	Symptoms	Solutions
Portal cannot access database	<p>The portal may not show any data or show an error message such as:</p> 	<p>There are a number of things to check, including:</p> <ul style="list-style-type: none">- Ensure the database is running- Check the configuration of the portal (WEB-INF/web.xml)- Ensure that both the portal server can access the database server (connection issues)

Issue	Symptoms	Solutions
<p>Portal hangs on login</p>	<p>After the username and password have been entered the main page may not load for a while. There should be an indication in the bottom left corner (see figure below) to show that progress is being made on the initial loading.</p>  <p>If this count stalls there may be issues relating to the initial page.</p>	<p>There are a few possible causes and solutions, including:</p> <ul style="list-style-type: none"> - Check the Google map key found in JndmsPortal.html. This is the most likely cause as an invalid key will prevent any portion of the initial page from loading. - Ensure that access to the configured map servers is available. If the map servers are off line or quite slow the initial page may stall. - Check the connection to the database.
<p>Events are not sent to JSS</p>	<p>You may get connection errors when using the jss_client.jar.</p>	<p>These are most often caused by configuration errors, especially in the endpoint parameter. Ensure that the endpoints match up to the server being used.</p>

Issue	Symptoms	Solutions
Log errors about port in use.	When examining the web application server logs you may see errors related to “cannot bind to port” or that the port is already in use.	This generally indicates that something is already running on the port required. This may be because another web server or web application server has been installed or that a previous instance is still running. You should ensure that nothing else is running and may have to examine the task manager.
Log errors about database connections.	When examining the web application server logs you may see errors relating to database connections. These are generally in the form of Java exceptions and may indicate that a connection cannot be made or that the username or password is invalid.	You should check the database to ensure that it is running and check the configuration of the JSS (WEB-INF/jss.properties).

Issue	Symptoms	Solutions
<p>Web application server may take a long time to start.</p>	<p>After running startup.bat it may take a while for the application server to start serving requests.</p>	<p>You may want to remove the logs before running. The logs can be quite verbose and start to impede performance after a while.</p> <p>The web application server logs may also show an indication of what is happening during this time. Often errors are being reported.</p>

5.3 Database Maintenance

The database is central to much of JNDMS and may be the cause of many issues. Oracle provides a tool called Enterprise Manager that should be used to view the status of the database and perform any routine maintenance.

5.3.1 Enterprise Manager

The Enterprise Manager should be running on the database server when the database is active. It can generally be found at:

<https://servername:1158/em>

On Linux server you can perform the following:

- Check status of enterprise manager:
 `$ORACLE_HOME/bin/emctl status agent`
 This should respond with "Agent is Running and Ready"
- To start enterprise manager (if not running), log in as oracle and run:
 `$ORACLE_HOME/bin/emctl start dbconsole`
- To check the URL to access the enterprise manager:
 `$ORACLE_HOME/bin/emctl status dbconsole`

On Windows installations entries to the Enterprise Manager for each database should have been added to the start menu.

Most actions for the enterprise manager can be done with the oracle 'system' account however some actions may require the 'sysdba' role. The Enterprise Manager will inform when different credentials are required.

Common tasks such as ensuring the database is running, checking the current system load, checking how large the database is or to see if there are any issues with available space can all be done through the web portal.

5.3.2 JNDMS Datasets

JNDMS depends on valid data to perform and a number of datasets have been created for demonstration purposes. Most of these datasets exist as Oracle export dump files. These dump files can be loaded into a schema using Oracle tools.

There are several preset datasets including:

- Stage1.dmp. This provides only basic lookup tables.
- Stage8.dmp. This provides a complete simulated environment but does not include vulnerability data.
- Vuln1.dmp. This is a complete simulated environment, including vulnerability information.
- Iat1-stage1.dmp. This is a base for IAT data.

To load the dataset you must select a schema that the JSS and Portal have been configured to use. You must also know the schema that was used to create the dump file. The old schema information can be found by either loading the dump file into the Enterprise Manager or by examining the text file that corresponds to the dump file found in the code repository. To import data:

- `impdp [user]/[password]@[oracle service] DIRECTORY=[data pump dir] DUMPFILE=stage8.dmp REMAP_SCHEMA=[old schema]:[new schema] LOGFILE=imjport.log`
- Where:
 - ♦ User, password and oracle service are database connection parameters.
 - ♦ The data pump directory is a server configured value in which the dump files must reside. Valid values for your database can be found using the Enterprise Manager under the Schema tab. The dump files may have to be copied from the code repository to the data pump directory if they are not already there.
 - ♦ The old and new schema names reference what named Oracle schema will be used to load the data under.

Ant targets for the above actions can also be found in the JDW project in the code repository (examine build.xml for details).

Another common action for datasets is when two commonly used datasets are preconfigured and you want to swap between them. An example batch file named swapdb.bat (found in c:\jndms\liferay\bin) has been created to swap between a common JNDMS schema and a common IAT schema. All of the configuration and schemas must be setup to allow this swap to occur.

You can also manually perform the swap that is done by the swapdb.bat by editing the configuration files for the portal as well as the portal. You should examine the configurable values found in the portal (WEB-INF/web.xml) and the core services (WEB-INF/jss.properties). See the 'database' paragraph under the section 5.2.

List of symbols/abbreviations/acronyms/initialisms

ACL	Access Control List
AJAX	Asynchronous JavaScript and XML
AM	Asset Management
API	Application Program Interface
BID	Bugtraq ID. This tracks vulnerabilities reported through the Bugtraq mailing list.
BPS	Boundary Protection System
BRE	Business Rules Expert
C2	Command and Control
C2IEDM	Command and Control Information Exchange Data Model
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance
CA	Computer Associates
CANUS	Canadian and US
CAP	Common Alerting Protocol
CAPI	Cryptographic Application Programming Interface
CDRL	Contract Data Requirements List
CFNOC	Canadian Forces Network Operations Centre
CIA	Confidentiality, Integrity and Availability
CIK	Crypto Ignition Key

CIRT	Computer Incident Response Team
CMDB	Configuration Management Database
CME	Common Malware Enumeration
CND	Computer Network Defense
CNES	Canadian Network Encryption System
CO	Commanding Officer
CONOPS	Concept of Operations
CoS	Class of Service
COTS	Commercial Off The Shelf
CSE	Communications Security Establishment
CVE	Common Vulnerability Exposures
CVSS	Common Vulnerability Scoring System
DHCP	Dynamic Host Configuration Protocol
DID	Data Item Description
DMF	Device Modeling Framework
DMFD	Device Modeling Framework Definition
DND	Department of National Defence
DRDC	Defence Research & Development Canada
DRDKIM	Director Research and Development Knowledge and Information Management

DREnet	Defence Research Establishment Network
DSS	Decision Support System. Part of JNDMS
DVPNI	Defence Virtual Privet Network Infrastructure
EAL	Evaluation Assurance Level. These levels are defined by the Common Criteria guidelines.
EIM	Enterprise Information Management. Part of JNDMS
ETL	Extract, Transform, Load
GIS	Geographic Information System
GUI	Graphical User Interface
GWT	Google Web Toolkit
HIDS	Host Intrusion Detection System
HIPS	Host Intrusion Prevention System
HTML	Hypertext Markup Language
HTTP	Hyper Text Transfer Protocol
I&A	Identification and Authentication
IAT	Impact Assessment Tool
IATF	Information Assurance Technical Framework
ICMP	Internet Control Message Protocol
IDE	Integrated Development Environment
IDS	Intrusion Detection Systems

INE	In-line Network Encryptor
IP	Internet Protocol
IPSec	Internet Protocol Security
ISM	Intellitactics Security Manager
ISP	Internet Service Provider
IT	Information Technology
ITI	Information Technology Infrastructure
JAR	Java ARchive. This is an archive file format defined by Java standards.
J2EE	Java 2 Enterprise Edition
JDBC	Java Database Connectivity
JDW	Jndms Data Warehouse
JNDMS	Joint Network and Defence Management System
JNDMS	Joint Network Defence and Management System
JSR	Java Specification Request
JSS	JNDMS System Services
JUI	JNDMS User Interface
KMI	Key Management Infrastructure
KML	Keyhole Markup Language - A GIS format defined by Google
LMP	Link Management Protocol

MARLANT	Maritime Forces Atlantic
MARPAC	Maritime Forces Pacific
MCOIN	Maritime Command Operation Information Network
MDB	Management Database. This refers to the datastore used by the CA products.
MTTRS	Mean Time To Restore Service
MUX	Multiplexer
NATO	North Atlantic Treaty Organization
NDHQ	National Defence Headquarters
NIAC	National Infrastructure Advisory Council
NIC	Network Interface Card
NIO	Network Information Operations
NIST	National Institute of Standards and Technology
NSM	Network Systems Management. This is part of the Unicenter product line.
NTSM	National Telecommunication Management System
NVD	National Vulnerability Database
ODB	Operations Database
ODBC	Open Database Connectivity
OOB	Out Of Band
OODA	Observe, Orient, Decide, Act

OpenGIS	Open Geodata Interoperability Specification
OSVDB	Open Source Vulnerability Database
PKI	Public Key Infrastructure
POC	Point of Contact
PWGSC	Public Works and Government Service Canada
QoS	Quality of Service
R&D	Research & Development
RDBMS	Relational Database Management System
RDEP	Remote Data Exchange Protocol
RFC	Request For Comments (Internet Standards documents)
RFC	Request For Change. A formal request for change on a network within DND.
RSS	Real Simple Syndication
SA	Situational Awareness
SCC	Security Command Centre
SCEM	Secure Common Email
SCI	Special Compartmented Information
SCP	Secure CoPy
SDA	Service Delivery Area
SDNS	Secure Data Network System

SDP	Service Delivery Point
SDW	Security Data Warehouse
SIM	Security Information Management. Part of JNDMS
SIP	Service Interface Point
SML	Strength of Mechanisms Level
SMS	Systems Management Server
SMTP	Simple Mail Transfer Protocol
SNI	Secure Network Infrastructure
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SRA	Secure Remote Access
SSH	Secure Socket Shell
SSL	Secure Sockets Layer
TBD	To Be Determined
TCP	Transmission Control Protocol
TD	Technology Demonstrator
TDP	Technology Demonstration Project
TTP	Trusted Third Party
UDP	User Datagram Protocol

UPS	Uninterrupted Power Supply
VA	Vulnerability Assessment
VPN	Virtual Private Network
WAN	Wide Area Network
WAR	Web application ARchive. A format defined by Java standards for deploying web applications.
WGS 84	World Geodetic System 1984
WSDP	Web Services Developer Pack
XML	eXtensible Markup Language
XSLT	extensible Stylesheet Language Transformations

